NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA

THESIS

APPLICABILITY OF SEI'S CAPABILITY MATURITY MODEL IN JOINT INFORMATION TECHNOLOGY, SUPREME COMMAND HEADQUARTERS

by

Jitti Thongmuang

June, 1995

Principal Advisor:	William R. Gates
Associate Advisor:	Lee Edwards

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
June 1995

3. REPORT TYPE AND DATES COVERED
Master's Thesis

4. TITLE AND SUBTITLE Applicability of SEI's Capability Maturity Model in Joint Information Technology, Supreme Command Headquarters

5. FUNDING NUMBERS

6. AUTHOR(S) Jitti Thongmuang

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)*

The Software Engineering Institute's (SEI) Capability Maturity Model (CMM) is analyzed to identify its technological and economic applicability for the Joint Information Technology (JIT), Supreme Command Headquarters, Royal Thai Ministry of Defense. Kurt Lewin's force field theory was used to analyze different dimensions of CMM's applicability for JIT's organizational environment (defined by the stakeholder concept). It suggests that introducing CMM technology into JIT is unwarranted at this time. CMM fails to incorporate several dimensions of software engineering management which are particularly relevant to JIT (e.g., human resource management, software engineering methods, tools and technology constraints and the model's applicability to a small organization). Furthermore, JIT faces technological and cultural barriers (e.g., little enthusiasm among JIT's personnel for accommodating new technology and change, insufficient budgetary resources, a short-term decision time horizon which may under value the long-term benefits of software process improvement, and incongruity between JIT's hierarchial-bureaucratic management and CMM's management philosophies). However, this study may help germinate the interest of JIT's technology gatekeeper in adopting CMM in the near term.

14. SUBJECT TERMS Capability Maturity Model, software engineering, software process, technology adopting, stakeholder and force field.

15. NUMBER OF PAGES 88

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
Unclassified

18. SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

19. SECURITY CLASSIFICATION OF ABSTRACT
Unclassified

20. LIMITATION OF ABSTRACT
UL

Approved for public release; distribution is unlimited.

APPLICABILITY OF SEI'S CAPABILITY MATURITY MODEL IN JOINT INFORMATION TECHNOLOGY, SUPREME COMMAND HEADQUARTERS

Jitti Thongmuang
Major, Royal Thai Army
B.S., Norwich University, 1981

Submitted in partial fulfillment
of the requirements for the degree of

**MASTER OF SCIENCE IN
INTERNATIONAL RESOURCE PLANNING AND MANAGEMENT**

from the

**NAVAL POSTGRADUATE SCHOOL
June 1995**

Author: ______________________
Jitti Thongmuang

Approved by: William R. Gates
William R. Gates, Principal Advisor

____
Lee Edwards, Associate Advisor

____
David R. Whipple, Chairman
Department of Systems Management

ABSTRACT

The Software Engineering Institute's (SEI) Capability Maturity Model (CMM) is analyzed to identify its technological and economic applicability for the Joint Information Technology (JIT), Supreme Command Headquarters, Royal Thai Ministry of Defense. Kurt Lewin's force field theory was used to analyze different dimensions of CMM's applicability for JIT's organizational environment (defined by the stakeholder concept). It suggests that introducing CMM technology into JIT is unwarranted at this time. CMM fails to incorporate several dimensions of software engineering management which are particularly relevant to JIT (e.g., human resource management, software engineering methods, tools and technology constraints and the model's applicability to a small organization). Furthermore, JIT faces technological and cultural barriers (e.g., little enthusiasm among JIT's personnel for accommodating new technology and change, insufficient budgetary resources, a short-term decision time horizon which may under value the long-term benefits of software process improvement, and incongruity between JIT's hierarchial-bureaucratic management and CMM's management philosophies). However, this study may help germinate the interest of JIT's technology gatekeeper in adopting CMM in the near term.

TABLE OF CONTENTS

I. INTRODUCTION

A. BACKGROUND

Joint Information Technology (JIT) Supreme Command Headquarters is highly regarded as the Royal Thai Ministry of Defense's premier organization in the field of software engineering technology. JIT employs more than two hundred information technology professionals, one-fifth of which are software practitioners. Their mission is to develop and provide a common or multiservice software that can be transparently used by all defense organizations.

In the recent past, there has been a growing demand on computer hardware and software in defense organizations. The recognition that information technology can effectively improve defense activities is an underlining cause. It is quite common for each Service (e.g., Army, Navy, and Air Force) to seek out their own information technology solutions based on their requirements and program funding. Compatibility and interoperability have become central issues for the Supreme Command Headquarters. It must coordination these vast and varied resources to coherently unify command, control, communication and intelligence across Services. For that reason, JIT 's mission is to act as an information technology integrator for the Royal Thai Ministry of Defense.

From JIT's past experience, software development problems are common. These technological-managerial problems lead to program schedule slippage, cost overruns, and delivered software that does not meet the stated requirements. These traditional software management shortfalls, including the shrinking budget and the difficulty of recruiting and retaining software engineering personnel has made JIT's software engineering mission a Herculean task.

Because of similar considerations, the US Department of Defense referred to software problems as its "achilles heel" (Kitfield, 1989, p.1). Since software production is labor intensive, the US Department of Defense worried that there would not be enough qualified labor to produce critical software (Raguindin, 1994, p.2). To circumvent these perplexing problems, DoD established the Software Engineering Institute (SEI) to "bring the DoD Services and Agencies the best available tools and techniques for the efficient design and production of reliable and adaptable software systems" (DoD Joint Service Task Force, 1983, p.1-2); and to "promote the evolution of software engineering from an ad-hoc, labor-intensive activity to a discipline that is well managed and supported by technology" (Software Engineering Institute, 1994).

In assessing and evaluating a software organization, SEI developed the Capability Maturity Model (CMM) and Maturity Questionnaire as diagnostic tools for probing organizational software development activities. SEI based its CMM's concepts on Total Quality Management (TQL) and software process paradigms. In simple terms, CMM proposes that the quality of a product is directly related to the process with which it is produced. Continuous process improvement reduces variance or "noise" in software development, as evidenced by schedule delays, cost overruns and technical shortfalls. The frequency of meeting customer requirements increases (Paulk et al, 1993, p.23). More importantly, process improvement normally leads to an increase in productivity (Herman and Lewis, 1993, p.6). The implication is that focusing attention on the process used to develop software products effectively addresses the problems facing software development. For that reason, CMM has become a de facto standard by which the US Department of Defense assesses its software organizations and evaluates its software contractors (Joyce, 1994, p.56).

On the whole, field experience in implementing CMM indicates the outcomes have been satisfactory and productive. Among the successful examples are the source selection for awarding a $95 million software avionics contract at Naval Weapons Center in China Lake (Rugg, 1993, p.36); the SEI's rating of maturity Level III for both Sacramento Air Logistics Center's Software Engineering Division (Joyce, 1994, p.53) and the Army's Fort Lee Center (Joyce, 1994, p.57); and the SEI's rating of maturity Level V, which is the SEI's highest level, for Motorola India Electronic's Bangalore Software Center (Sims, 1994, p.92).

At this stage, it may be reasonable for JIT to acquire CMM, hoping that it will prove as successful for the Royal Thai Ministry of Defense as it has for the US Department of Defense. However, the benefits of CMM can not be judged on their face value. Its application in JIT must be systematically and logically analyzed to ensure the full potential benefits of this technology.

B. OBJECTIVES

The objective of this thesis is to provide critical information to help Joint Information Technology (JIT) Supreme Command Headquarters develop an appropriate policy toward acquiring and adopting SEI's CMM. To meet this objective, an overview of CMM, including its field experience, will be presented. Both technological and economic attributes of the model will be investigated and analyzed as they pertain to JIT's organizational environment.

C. THE RESEARCH QUESTIONS

The primary research question for this thesis is:

- Is the Capability Maturity Model applicable in the Royal Thai Ministry of Defense's Joint Information Technology?

To support the primary research question, this thesis will address the following subsidiary questions:

1. What is the Capability Maturity Model?

2. What is the Joint Information Technology's current software engineering technology?

3. What criteria should be used to assess the Capability Maturity Model's applicability within the Joint Information Technology's organizational environment?

4. What approaches should be used in analyzing the Joint Information Technology's organizational attitude/cultural readiness for introducing the Capability Maturity Model?

D. SCOPE AND LIMITATIONS

This thesis is a case study of the Software Engineering Institute's Capability Maturity Model. It focuses on CMM's applicability for Joint Information Technology, Supreme Command Headquarters. Evaluating CMM's technical capability is beyond the scope of this thesis; only the general attributes of the model and its practices are presented. Further, CMM's technical detail will be presented sufficiently to support the main focus of this study.

CMM's applicability attributes were primarily derived from SEI's publications. Software process improvement appraisals are only released with the consent of organizations involved. Beneficial results of the model are commonly reported. There is no definitive information available which contradicts these productive outcomes. In addition, JIT's organizational characteristic were assessed by interviewing selected personnel, not the entire organization.

E. RESEARCH METHODOLOGY

The research for this thesis was conducted in two steps. The first step involved an intensive literary search for published material concerning the Software Engineering Institute and its Capability Maturity Model. The main effort focused on describing CMM's applicability attributes that are pertinent for JIT's organizational environment. Moreover, inputs from members of Naval Postgraduate School faculty were synthesized to develop an applicability model.

The second part of this research involved interviews to obtain JIT's organizational attitudes and readiness for introducing CMM. Senior executive officers of JIT were interviewed about the potential for transferring software engineering tools and technology. Personnel from the Software Development Division were interviewed about their respective software engineering practices.

F. ORGANIZATION OF STUDY

The remaining chapters of this thesis are organized into three major parts. The first part, consisting of Chapters II and III, provides general information about SEI's CMM and JIT. Chapter II describes CMM. Chapter III describes JIT's organizational structure and its software development activities.

The second part, Chapter IV, develops applicability attributes both for CMM and JIT. The interactions between the two are thoroughly investigated.

Finally, Chapter V summarizes the applicability analysis. This provides preliminary information about whether JIT should acquire and adopt CMM.

II. SOFTWARE ENGINEERING INSTITUTE'S CAPABILITY MATURITY MODEL

An explanatory overview of the Software Engineering Institute's (SEI) Capability Maturity Model (CMM) will be provided in this chapter, including SEI's methodology for software engineering project management and a description of the Capability Maturity Model (CMM) framework. Finally, the operations and uses of CMM will be briefly discussed.

A. SOFTWARE PROCESS VIEW

The software process paradigm rests on the precept that to improve the productivity of software organizations and the quality of software products, efforts should focus on the software manufacturing process itself. There is a presumption that focusing quality improvement efforts on software products will subsequently reduce the over all life cycle cost (Dowson, 1993, p.55). Specifically, improving the software process is expected to achieve the following desirable objectives:

- Software projects will be more effective: the resources will be used more efficiently so software products will take less effort to produce.
- Software projects will be more predictable: it will be possible to estimate the resources and time needed to produce a product with greater accuracy.
- Software products will have higher quality (Dowson, 1993, p.55).

A software process can be defined as "a set of activities that begins with the identification of a need and concludes with the retirement of a product that satisfies the need; or more completely, as a set of activities, methods, practices,and transformations that people use to develop and maintain software and its associated products (e.g., project plans, design documents, codes, test cases and user manuals" (Werth, 1993, p.8;Paulk et al, 1993, p.20). An exemplar of this is the waterfall model (Figure 1), which begins with the system requirements and proceeds to an analysis and design system specification. It continues further with coding, testing, and implementing the system as well as operating and maintaining the system.

B. SOFTWARE PROCESS MATURITY MODEL DEVELOPMENT

Recognizing the urgent demand for coping with the so-called "software crisis" -- cost overruns, late deliveries, poor reliability and user's dissatisfaction (Abdel-Hamid and Madnick, 1991, p.6), in 1982 the U.S. Department of Defense (DoD) formed a joint service task force to review these software symptoms and deficiencies. The results were reflected in several initiatives. Among the most significant were establishing the Software Engineering Institute (SEI) at Carnegie-Mellon University, the Software Technology for Adaptable Reliable Systems (STARS) Program, and the Ada Program.

SEI is under a contractual relationship with DoD as a Federally Funded Research and Development Center (FFRDC) (Software Engineering Institute, 1994). Its purpose is to "bring the DoD Service and Agencies the best available tools and techniques for the efficient design and production for reliable and adaptable software" (DoD Joint Service

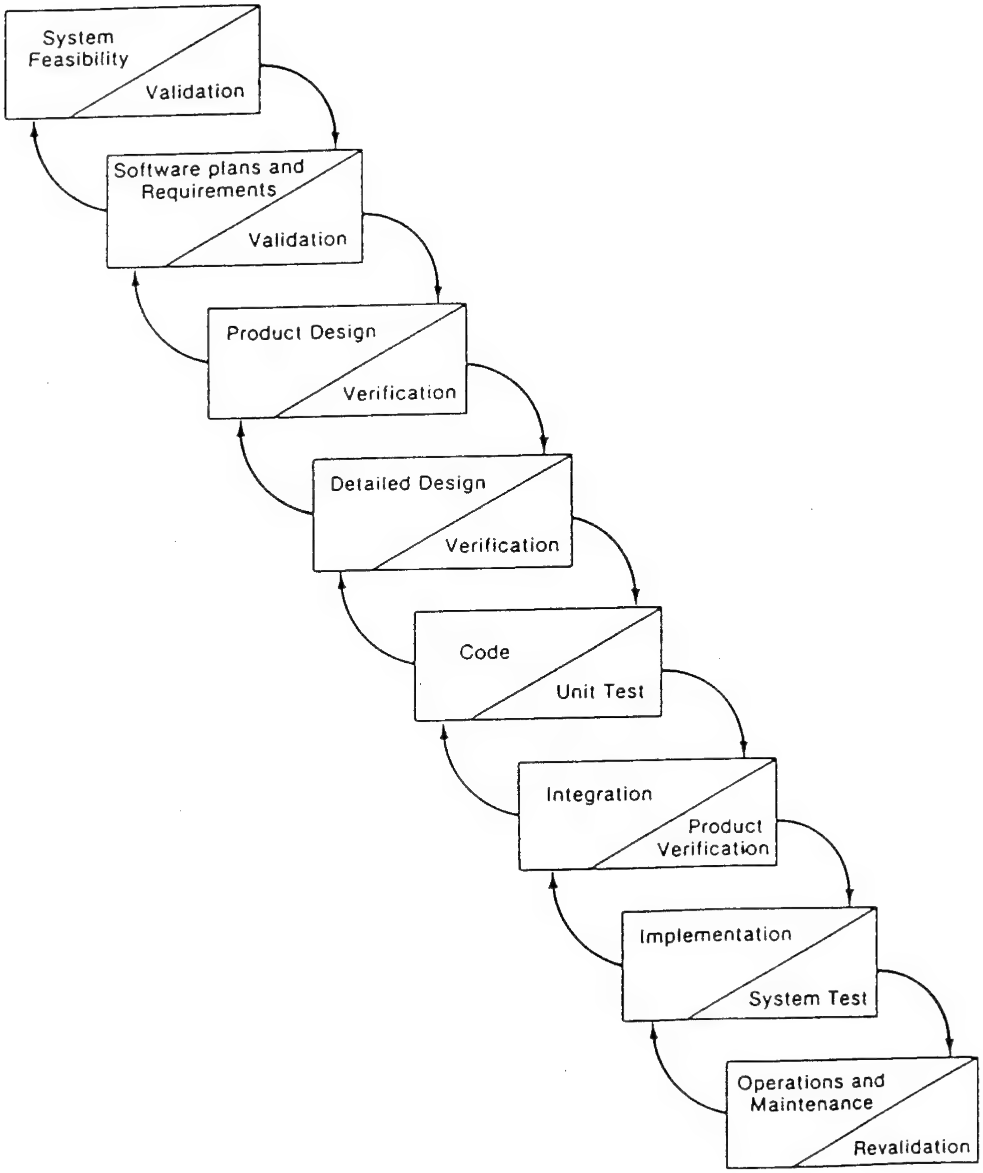

Figure 1 The Waterfall Model for the Software Life Cycle

(Abdel-Hamid and Madnick, 1991, p.45)

Task Force, 1983, p.1-2). In striving to achieve the goal, SEI maintains a state-of-art software development environment that promotes the evolution of software engineering from an ad-hoc, labor-intensive activity to a discipline that is well managed and supported by current technology (Software Engineering Institute, 1994). Presently, SEI concentrates its resources on four primary technical areas: 1) software process 2) software risk management 3) product attribute engineering, and 4) software engineering techniques (Software Engineering Institute, 1994). Of these, the Capability Maturity Model (CMM) is the most publicized and best known project in the software engineering community. The model has become "an industry yardstick" (Software Engineering Institute, 1994).

The SEI's CMM was motivated "by the increasing importance of software in DoD procurement and the need of all the [US military] services to more effectively evaluate the ability of their software contractors to competently perform on software engineering contracts" (Bamford and Deibler II, 1993, p.68). Indeed, it was the combined effort of the US Air Force, the MITRE Corporation, and particularly SEI, to seek a technically sound and consistent method for the acquisition community to identify the most capable software contractors (Humphrey, 1992, p.9). This resulted in a questionnaire and a framework for evaluating software organizations according to the maturity of their software process. The maturity questionnaire, which helps to identify the organizations' software process strengths and weaknesses, encompasses three principle areas in software engineering: organization and resource management, software engineering process and its management tools and technology (Humphrey, 1992, p.9). Moreover, the basic ideas behind this technical research came from several sources such as Deming's statistical process control principles, Shewhart's Plan-Act-Check-Do process improvement cycle, and Crosby's Quality model (Humphrey, 1992, p.10). In short, the maturity model is based on the premise that "the quality of a software product stems, in large part, from the quality of the process used to create and maintain it" (Humphrey and Sweet, 1987, p.10). In addition, software engineering practice must incorporate the notion that "the process of producing and evolving software can be defined, managed, measured, and progressively improved" (Humphrey and Sweet, 1987, p.10).

The software process maturity model and SEI's assessment methods have been reviewed by individuals and organizations with a great deal of experience in software development and acquisition. They evolved and have been modified to be relevant to the dynamism of software engineering technology. For that reason, the software process maturity model was elaborated upon and updated into the Capability Maturity Model (CMM) (Humphrey, 1992, p.10). The latest version of CMM software is 1.1, which was released in 1993. It incorporates feedback from the community using Version 1.0. SEI expects that Version 1.1 will remain in use until at least 1996 (Paulk et al, 1993, p.20). Further, the CMM project is active in the International Standards Organization's (ISO) Software Process Improvement and Capability dEtermination (SPICE) project (Software Engineering Institute, 1994). This participation will help CMM be recognized worldwide.

C. THE CAPABILITY MATURITY MODEL

1. Fundamental Concepts

The framework of the Capability Maturity Model (CMM) is based on the software process paradigm --to improve the quality of the software products, the software process must be incrementally and continuously improved and measured (Werth, 1993, p.1-2). By the same token, product quality or project success is directly related to the quality or maturity of the software process (Herman and Lewis, 1993, p.6).

Software process capability can be described as the "inherent ability of a process to produce planned results and a capable software process is characterized as mature" (Humphrey, 1992, p.1). In simple terms, capability is as a gauge to measure and predict the most likely outcome of the next project in which a software organization is involved (Paulk et al, 1993, p.20). Software maturity implies the potential for improvement in the software process and indicates "both the richness of an organization's software process and the consistency with which it is applied in projects throughout the organization" (Paulk et al, 1993, p.20).

CMM is a software engineering management approach. It assesses the software organization's capability to produce high quality products in a consistent and predictable manner (Humphrey, 1992, p.1). It also provides recommend changes and guides improvement efforts in managing software projects (Herman and Lewis, 1993, p.7).

2. Maturity Levels and their Behavioral Characterization

SEI places a software organization into one of five levels of software process maturity (Figure 2). At each level, the organization has an unique and distinct process (Humphrey, 1992, p.11). The more mature the activities performed, the higher the organization's level of maturity (Dickeroff and Sommers, 1992, p.29). Each level builds on the capabilities of the lower levels and

- represents an historical phase of evolution for a software organization,
- represents a reasonable measure of improvement to achieve from the prior level,
- suggests interim improvements, goals, and process measures, and
- makes obvious a set of immediate improvement priorities once an organization's status in the framework is known (Herman and Lewis, 1993, p.9).

At the Initial Level (Level 1), an organization is generally characterized as having an ad hoc, or possibly chaotic process. There are no formal management procedures, cost estimation, nor project planning or controlling. Management has very little insight into the key software deficiencies. The products being developed are often not on the target. If the projects do succeed, it is generally because of the heroic and dedicated efforts of talented programmers rather than the capability of the organization. A code-and-fix strategy is a common practice to which the organization reverts when facing a crisis (Humphrey, 1992, p.11;Dickeroff and Sommers, 1992, p.29).

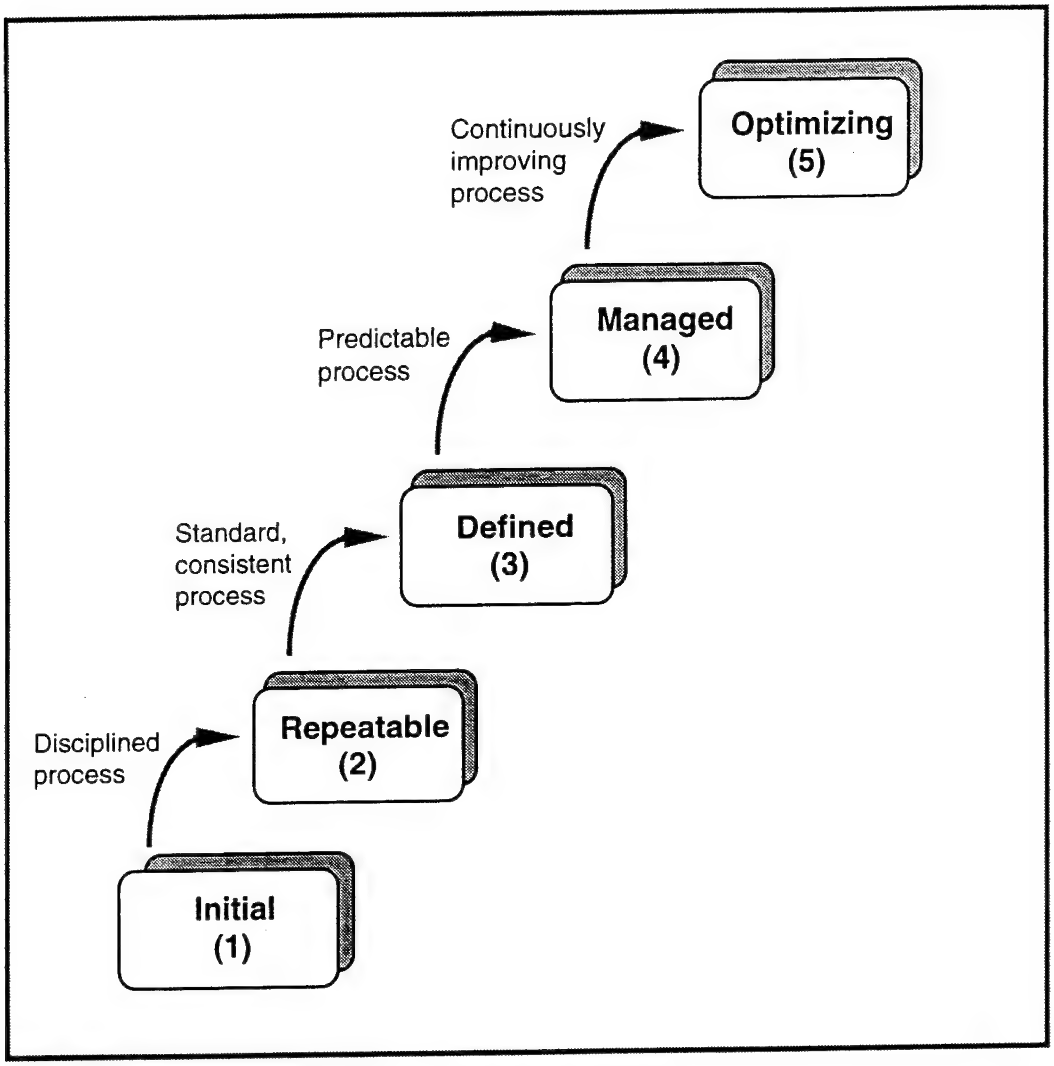

Figure 2 The Five Levels of Software Process Maturity

(Paulk et al, 1993, p.24)

To mature to the Repeatable Level (Level 2), an organization must institutionalize basic project controls. Project management needs to ensure proper control of resource allocation. Senior management must be exposed to and fully aware of the key software process problems and issues. Software quality assurance procedures must be established. Finally, the change control process needs to be formalized and institutionalized (Humphery, 1992, p.12;Dickeroff and Sommers, 1992, p.29-30).

An organization at the Repeatable Level (Level 2) is capable of repeating prior successes with similar projects. The major risk is posed when the organization faces the uncontrolled introduction of new technology and new challenges and problems. The prior experience and accumulated knowledge base used to estimate and predict project cost and completion time may no longer appropriate. Moreover, software quality and productivity are generally low because there is no orderly framework for improvement (Humphrey, 1992, p.11;Herman and Lewis, 1993, p.10).

In order to move from the Repeatable Level (Level 2) to the Defined Level (Level 3), an organization must define its standard software development process architecture. A Software Engineering Process Group (SEPG) must exist to "focus and lead the process improvement efforts, to keep management informed on the status of these efforts, and to facilitate the introduction of a family of software engineering methods and technologies" (Humphrey, 1992, p.11). Despite having a sound software development practice, there is little data to indicate the effectiveness of the defined process. Key challenges, including process measurement and data analysis, still remain (Herman and Lewis, 1993, p.11).

Advancing to the Managed Level (Level 4), an organization must collect quality and productivity measurements. An organization-wide process database, together with analysis tools, are needed. At this stage, the effectiveness of process improvement efforts are systematically determined. Consequently, management is able to shift its attention to areas with weaknesses, to ensure higher quality products (Dickeroff and Sommers, 1992, p.31).

To progress to the Optimizing Level (Level 5), the highest maturity level, management must support and implement automatic data collection and "redirect its resources from the [software] product to [the software development] process analysis and improvement" (Herman and Lewis, 1993, p.11). Automatic data collection reduces bias and the error inherent in human collection (as practiced in Level 4). An additional key activity at this level is "rigorous defect cause analysis and defect prevention" (Humphrey, 1992, p.11). Finally, it is imperative to use all available data to continuously optimize process development.

3. CMM and its Components

As stated before, the CMM framework both describes the characteristics of a mature software process and represents an evolutionary path for improving software processes into a well-managed, mature process (Werth, 1993, p.8). The internal structure of CMM is shown is Figure 3. Each maturity level, except Level 1, consists of several key process areas. Each key process area is grouped into five sections, called common

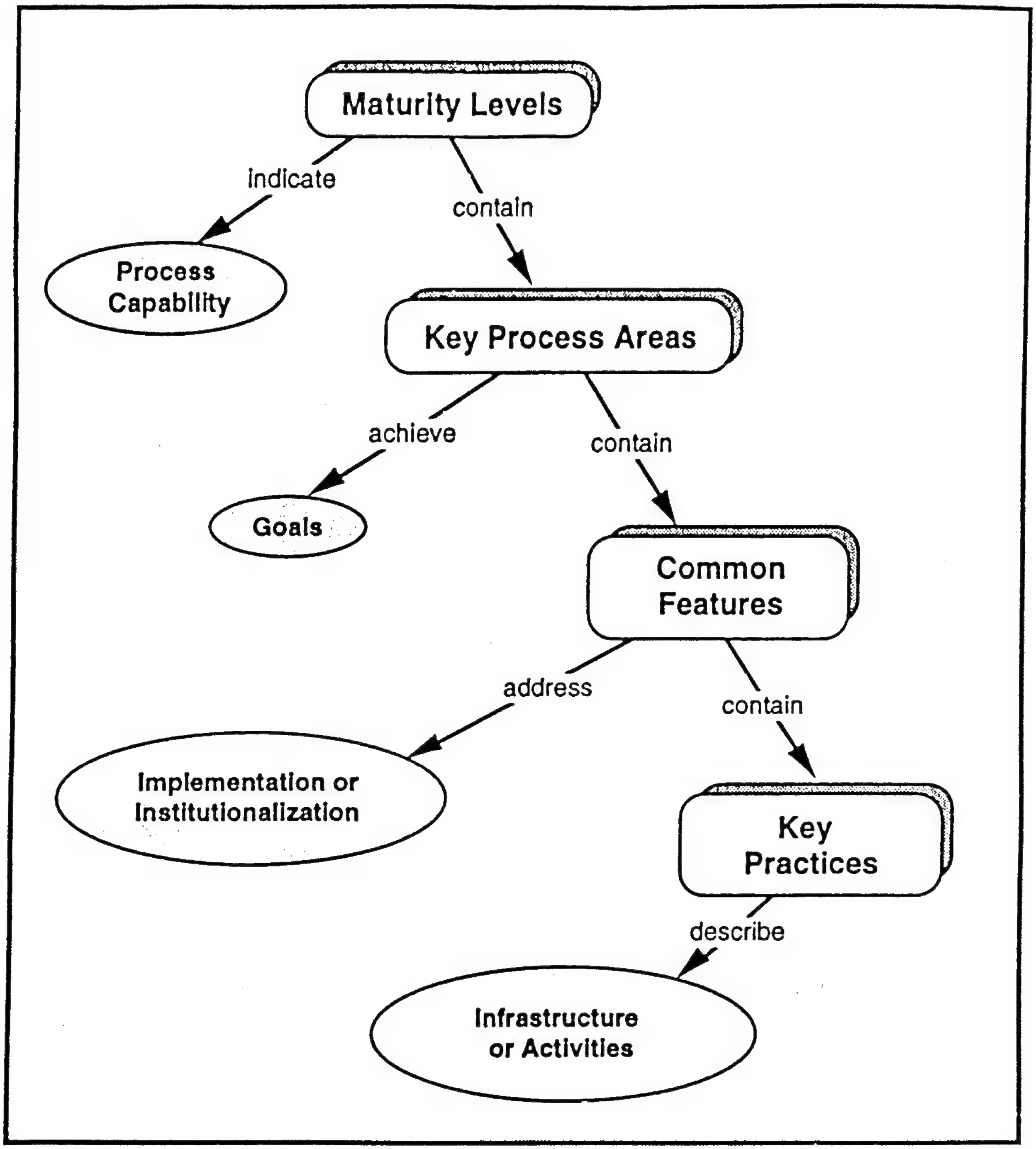

Figure 3 Capability Maturity Model Structure

(Werth, 1993, p.9)

features. The common features specify key practices (Paulk et al, 1993, p.24). These major components will be described in more detail below.

a. Maturity Levels

Each maturity level indicates a certain software process capability. It also describes how the software organization is expected to perform: initial, repeatable, defined, managed or optimizing. In fact, process capabilities predict the organization's expected results in managing the next software project based on its current process capabilities (Werth, 1993, p.9).

b. Key Process Areas

Key process areas (KPAs) identify areas where an organization should focus to improve its software development process. They pinpoint the most important issues that need to be addressed to reach different maturity levels, as Table 1 illustrates. In other words, KPAs may be viewed as requirements for achieving different maturity levels. Note that there are no KPAs for the Level 1 (Paulk et al, 1993, p.24).

Each KPAs includes a "cluster of related activities which are call key practices that, when performed collectively, achieve a set of goals which is considered important for enhancing process capability" (Paulk et al, 1993, p.24). Moreover, goals are used to determine if an organization or project has adequately implemented a certain KPA (Werth, 1993, p.11). Finally, KPAs provide building blocks, or fundamental activities for an organization attempting to improve its software process; each KPA is unique to a single maturity level.

Maturity Levels	Key Process Areas
Level 2: Repeatable	-Requirements Management -Software Project Planning -Software Project Tracking&Oversight -Software Subcontract Management -Software Quality Assurance -Software Configuration Management
Level 3: Defined	-Organization Process Focus -Organization Process Definition -Training Program -Integrated Software Management -Software Product Engineering -Intergroup Coordination -Peer Reviews
Level 4: Managed	-Quantitative Process Management -Software Quality Management
Level 5: Optimized	-Defect Prevention -Technology Change Management -Process Change Management

Table 1 Key Process Areas by Maturity Levels

(Paulk et al, 1993, p.25)

c. Common Features

In a simple word, the practices that describe the KPAs are grouped according to their similarities --common features. There are five groups of common features: 1) commitment to perform, 2)ability to perform, 3)activities performed, 4) measurement and analysis, and 5) verifying implementation (Paulk et al, 1993, p.25). On the whole, common features are attributes that indicate whether the implementation of a KPA is "effective, repeatable, and lasting" (Paulk et al, 1993, p.25).

d. Key Practices

Key practices describe the specific details of CMM (Werth, 1993, p.12). Key practices are the policies, procedures, and activities that most significantly institutionalize and implement the KPAs (Werth, 1993, p.8). Indeed, a key practice is a working definition of a KPA. Key practices describe what to do, but they do not mandate how that practice should be performed. The SEI technical report, "Key Practices of the CMM, Version 1.1" (Paulk et al, 1993), elaborates the key practices that correspond to each maturity level and provides extensive definitions and guidance on interpreting key practices (Werth, 1993, p.12).

e. Maturity Questionnaire

The maturity questionnaire consists of questions about the software process that sample practices in each key process area. Specific questions relate to specific key practices (Zubrow et al, 1994;Werth, 1993, p.12).

4. CMM in Practice

Figure 4 illustrates the relationship between the major CMM components (maturity level, KPA, key practice, and question). At the Repeatable Level, software project planning is one of the KPAs, and one of the key practices in software engineering planning is to estimate project size is. Thus, a typical question in the maturity questionnaire might be, "Is there any formal procedure for estimating the software size?" (Werth, 1993, p.9).

D. USES OF CMM

There are two major purposes for applying CMM: for software process assessment (SPA) and for software capability evaluation (SCE). The assessment and evaluation methods are based upon the CMM framework and use the SEI maturity questionnaire. Together, the model and questionnaire provide a structural basis identifying the organization's key strengths and weaknesses. The significant difference between assessment and evaluation comes from the way the results are used. For assessment, the results form the basis for an action plan for organizational self-improvement. For an evaluation, the results will be used to develop an organizational risk profile, which will be further applied in the source selection process and contract monitoring (Humphrey, 1992, p.17). Table 2 highlights the key differences between these two methods.

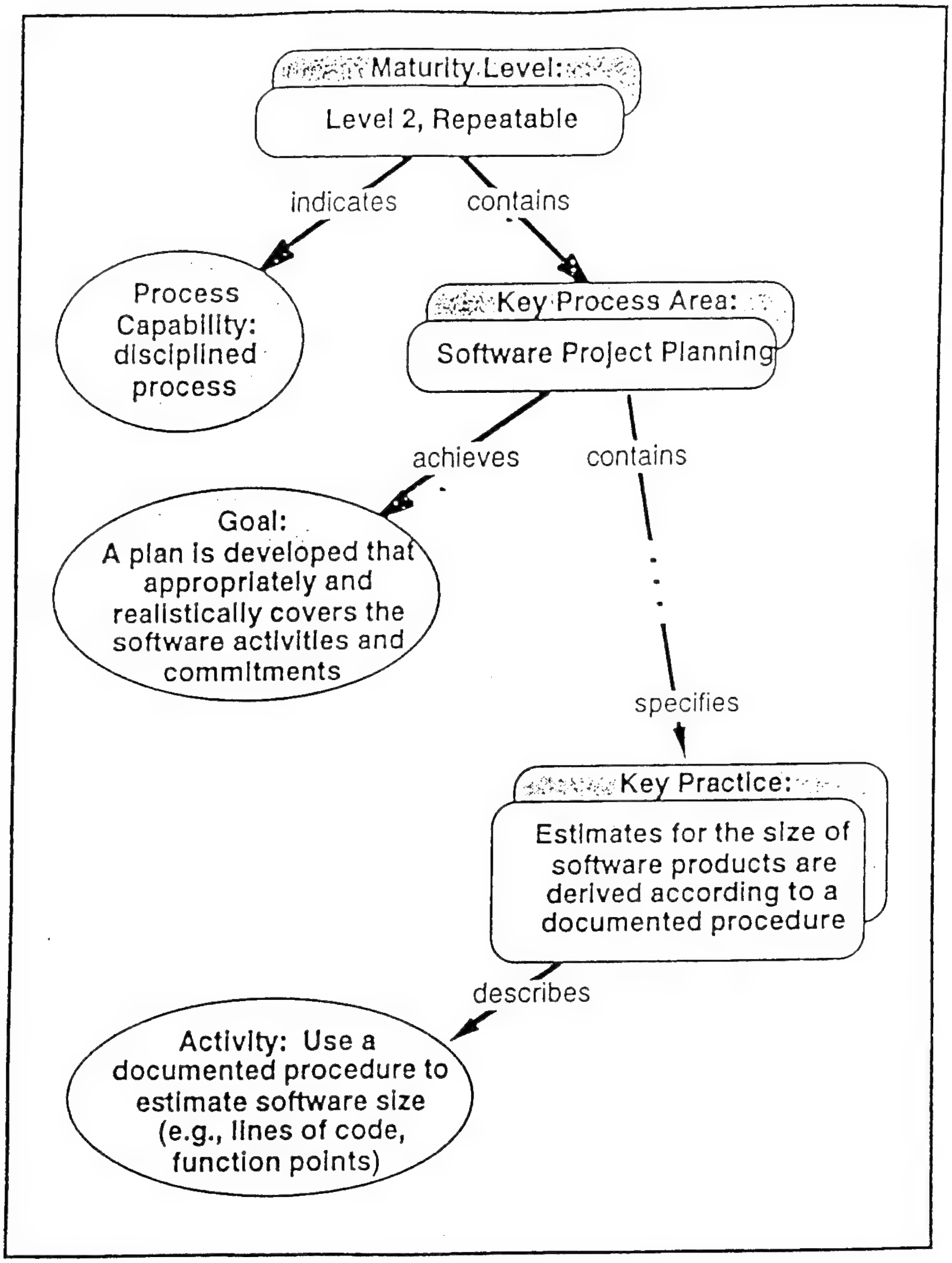
Maturity Level:
Level 2, Repeatable
indicates
contains
Process Capability: disciplined process
Key Process Area:
Software Project Planning
achieves
contains
Goal: A plan is developed that appropriately and realistically covers the software activities and commitments
specifies
Key Practice:
Estimates for the size of software products are derived according to a documented procedure
describes
Activity: Use a documented procedure to estimate software size (e.g., lines of code, function points)

Figure 4 Example of CMM Structure

(Werth, 1993, p.10)

SPA	SCE
Used by organization to improve software process	Used by acquisition organization for source selection and contract monitoring
Results to organization only	Results to the organization and the acquirer
Assesses current practice	Substantiates current practice
Acts as catalyst for process improvement	Assesses commitment to improve
Provides inputs to improvement action plan	Analyzes contract performance potential
Collaborative-organization members on team, with representative of licensed SPA associate or SEI	Independent evaluation-no organization members on team
Applies to organization overall, not individual projects	Applies to performance for a particular contract

Table 2 Comparison between SPA and SCE

(Werth, 1993, p.7)

1. Software Process Assessment

A software process assessment (SPA) is an internal tool for an organization to identify its strengths, weaknesses, existing improvement activities, and key areas for improvement. SPA helps an organization to determine the current state of its software process and to develop an action plan for improvement (Werth, 1993, p.7).

The assessment must start with the senior level management's commitment to support the software process improvement. Then, an organization should arrange for SEI to conduct an assessment. Management must provide the necessary funding for the operation. Since management is willing to fund the assessment, they should be willing to act upon the recommendations provided (Humphrey, 1992, p.17;Werth, 1993, p.7-9).

The next step is to select five or six projects on which the organization is currently working to serve as a representative sample of the organization's software development process. The project managers are then given a questionnaire by the assessment team, which consists of six-to-ten experienced software professionals. The questionnaire covers all areas of each project's development process. Once the questionnaire is completed, the assessment team begins a four day on site assessment (Humphrey, 1992, p.18;Werth, 1993, p.7-9).

The on-site assessment begins with a briefing to management covering the ground rules and schedule. At that point, management cooperation and support is highly encouraged. The assessment team then meets privately to review the questionnaire answers. The team interviews several individuals to clarify the questionnaire responses. If necessary, project leaders are also interviewed to gain full insight into the current

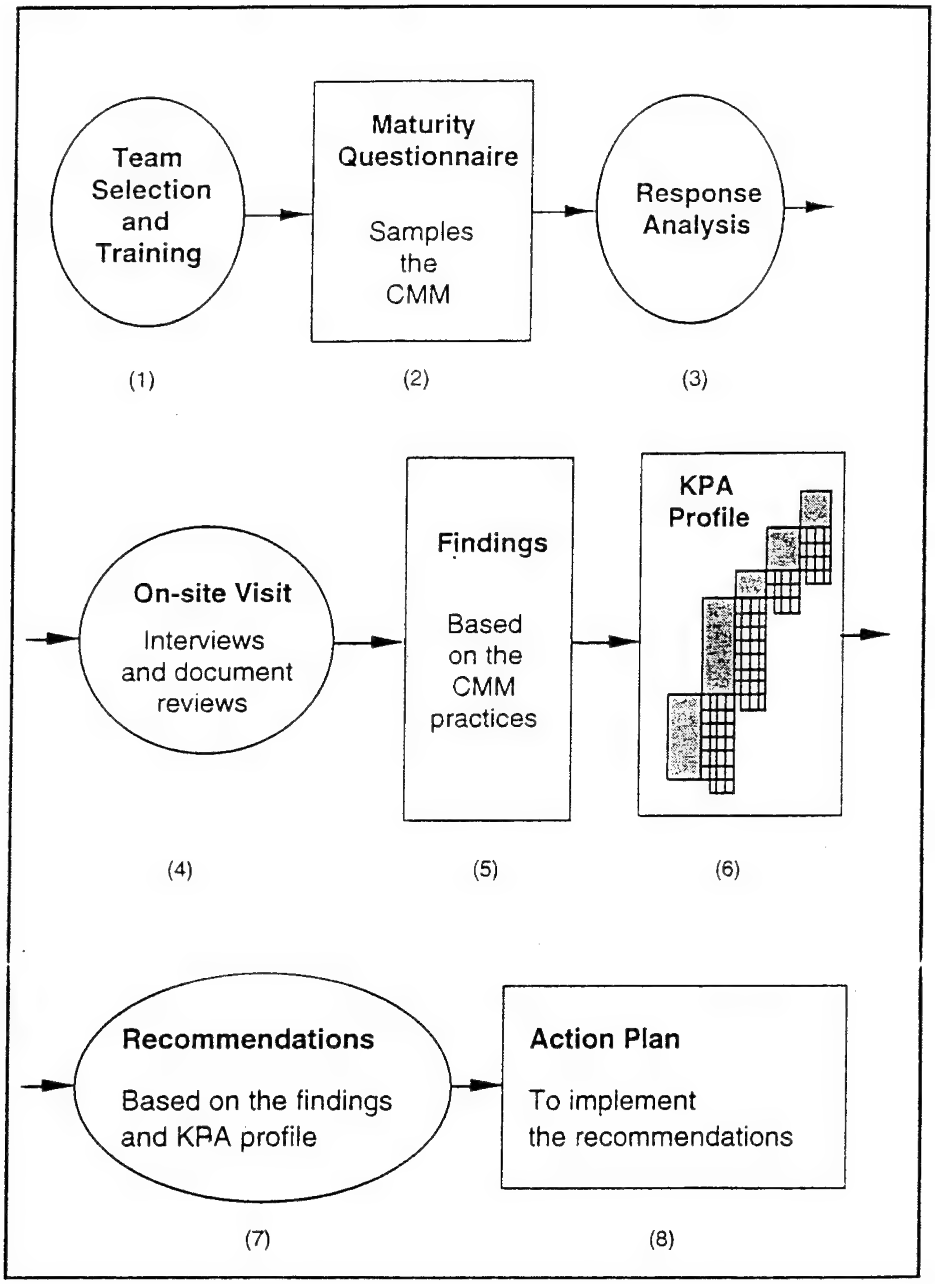
Team Selection and Training
(1)
Maturity Questionnaire
Samples the CMM
(2)
Response Analysis
(3)
On-site Visit
Interviews and document reviews
(4)
Findings
Based on the CMM practices
(5)
KPA Profile
(6)
Recommendations
Based on the findings and KPA profile
(7)
Action Plan
To implement the recommendations
(8)

Figure 5 Common Steps in SPAs and SCEs

(Werth, 1993, p.6)

development process. All these activities take the whole first day (Humphry, 1992, p.18;Werth, 1993, p.7-9).

The second day consists of discussions with technical individuals to provide further insight into the exact nature of the software development process. On the third day, the assessment team reviews the relevant documentation and might interview the individuals concerned as needed. At this stage, the team collects the most significant information concerning the software process. This completes the research portion of assessment. The findings are presented to the organizations's management on the following day (Humphrey, 1992, p.18-19;Werth, 1993, p.7-9).

The findings are presented in two forms: a briefing for senior management during the fourth day, and a written final report. The findings highlight the highest priority areas for process improvement. After the briefing and final report, the assessment team produces an action plan to the address the needed process improvements (Humphrey, 1992, p.18-19;Werth, 1993, p.9).

2. Software Capability Evaluation

A software capability evaluation (SCE) is an independent evaluation of an organization's software process as it relates to particular acquisition. External organizations (e.g., the DoD) use SCE to determine whether a particular software organization is capable of producing a high quality product on time and within budget. In fact, the method is used to indicate the risk associated with using that software supplier for a particular software acquisition (Werth, 1993, p.7). For that reason, an external organization, especially a government agency, applies SCE to gain insight into the organization's software development process capabilities.

The process begins when a government agency determines that an organization's software capabilities need to be evaluated. The government either sends its team to the SEI for training or hires an approved evaluation team. The team sends a questionnaire to the organization's management (the same questionnaire that is used in the SPA). The team then conducts a three day, on site evaluation during which they interview individuals to clarify answers and gain further insight into the process. All of the organization's responses must be documented. Consequently, the team spends a great deal of time reviewing the organization's documents and plans (Humphrey, 1992, p.19-20;Werth, 1993, p.10-11).

At the end of the third day, the team produces an evaluation report that assigns a maturity level to the organization and identifies the organization's strengths and weaknesses. The results of the evaluation become the property of the government agency and can be used in any manner desired. In short, the SPA is used to gauge the strengths and weaknesses of a given contractor's software development process (Dickeroff and Sommers, 1992, p.32).

E. IMPLICATIONS OF CMM

It is counter-productive for an organization to attempt to reach Level 5 without progressing through Level 2, 3, or 4. Each level builds on the prior levels. An organization can institute specific process improvements which may belong in higher levels at any time, but the stability of these improvements are at greater risk if they do not rest on a complete foundation (Paulk et al, 1993, p.24).

As an organization's maturity increases, three types of improvements in performance can be expected. First, the discrepancy between target results and actual results decreases across projects. Second, the variability of actual results around targeted results decreases. Third, cost and development time decrease while productivity and product quality increase (Paulk et al, 1993, p.23).

There is no information on how long it takes for an organization to progress from the Initial Level to the Optimized Level. Nevertheless, it requires high-level management support and long-term commitment. Management must take an active role in modifying the way software professionals and practitioners do their jobs and be willing to commit resources to support the transformation (Dickeroff and Sommers, 1992, p.31-32).

F. SUMMARY

The Software Engineering Institute (SEI) was as a response to the software crisis confronting DoD. Its mission is to provide leadership in software engineering to improve the quality of systems that depend on software. Of SEI's research accomplishments, the most visible and well-known is the Capability Maturity Model (CMM). The model is based on the notion that all quality improvement efforts should be focused on the software process, not on the software products. The software process paradigm, together with Deming's principles of statistical quality process control, Crosby's management of quality and other quality experts' premises form the core hierarchical structure of the CMM framework and its maturity questionnaire.

The basic components of CMM are the five levels of software process maturity (Initial, Repeatable, Defined, Managed and Optimizing), key process areas, key practices and questions. It is the key practices that provide the link between the CMM structure and the maturity questionnaire. As an organization matures, the difference between targeted results and actual results decreases, and development time and cost decrease. This increases productivity and quality of the software product.

CMM enables an organization to determine where it stands within the five tier rating system. The ratings are derived from two methods of evaluation: software process assessment (SPA) and software capability evaluation (SCE). The SPA is primarily applied during an in-house assessment, while the SCE is used by many government agencies in developing a risk profile to assess contractors during software acquisition.

SEI is constantly monitoring feedback from CMM and its maturity questionnaire. In other words, CMM is a living technical document that constantly evolves and improves. SEI anticipates that the current CMM, Version 1.1, which was released in 1993, will remain the baseline until at least 1996. This will help an organization to strike a realistic balance between the need for stability and the goal of continuous improvement. In addition, the CMM project is moving toward the international level. It actively participates in the International Standards Organization's (ISO) Software Process Improvement and Capability dEtermination (SPICE) project.

III. JOINT INFORMATION TECHNOLOGY AND ITS SOFTWARE ENGINEERING TECHNOLOGY

This chapter describes the role of Joint Information Technology (JIT) in the Royal Thai Ministry of Defense (RTMoD). Its mission and organizational structure are introduced. Finally, JIT's software engineering practices are presented.

A. JIT AND RTMoD

Speaking in terms of strategic defense requirements and force structure, the Royal Thai Ministry of Defense (RTMoD) consists of three separate and distinct armed Services: Royal Thai Army (RTA), Royal Thai Navy (RTN),and Royal Thai Air Force (RTAF). Supreme Command Headquarters coordinates these three vast and complex organizations in their totality. Structurally, Supreme Command Headquarters performs its coordinating function using a hierarchical organizational structure. Figure 6 reflects the organizational design.

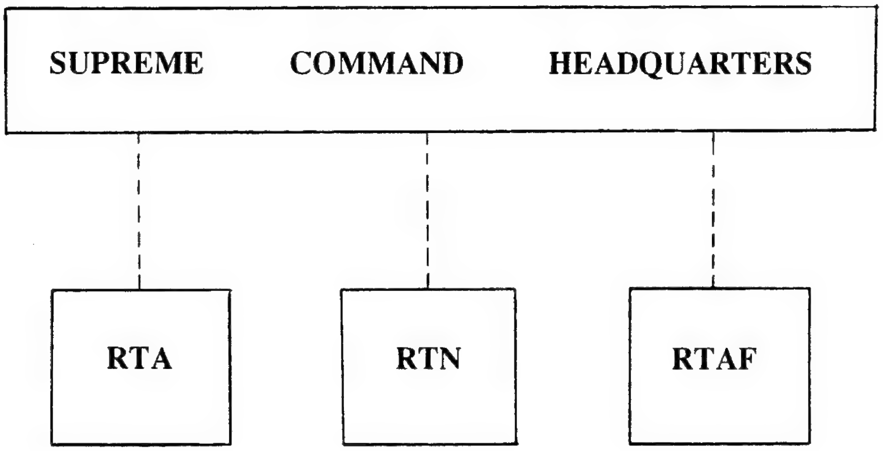

Figure 6 RTMoD Organizational Structure

The dashed line represents the functionality relationship between RTA, RTN, RTAF, and Supreme Command Headquarters. Note that the dashed line does not signify an authority relationship or chain of command. These three Services maintain their organizational autonomy.

To illustrate the functionality of Supreme Command Headquarters, consider how RTMoD deals with multi-service logistics operations. It is Joint Logistics, Supreme Command Headquarters which coordinates with Department of Army Logistics, Department of Navy Logistics and Department of Air Force Logistics. In a similar context, Joint Information Technology is responsible for jointly operating with the other Service's computing resource centers. The goal is to unify and integrate command, control, communication and intelligence. In fact, JIT's prime concern is establishing compatibility and interoperability across different computing resources platforms so that defense information can be shared.

In striving to increase its technological capability, JIT has already positioned itself as a telecommunication network hub and provides necessary information to the appropriate defense organizations (JIT's R&D Division, 1994). However, the network operations are at early stage. There are still major technical difficulties to overcome, especially in the areas of data communication, network management and network security. Table 3 shows the list of military installations that have been electronically linked with JIT. Different types of media are employed (e.g., cable, microwave, radio, fiber optics and satellites).

- Army Operations Center
- Navy Operations Center
- Air Forces Operations Center
- Burapa Task Forces
- Chantaburee-Trat Task Forces
- Army Data Processing Center
- Air Forces Logistics Center
- Joint Personnel
- Joint Intelligence
- Joint Operations&Training
- Joint Logistics
- Joint Comptroller
- Joint Communication
- First Army Region
- Second Army Region
- Third Army Region
- Fourth Army Region
- Third Infantry Division,TOC
- Sixth Infantry Division,TOC
- Army Engineering Command
- Air Forces Engineering Command
- Army Territorial Dept.
- Defense Intelligence Center
- Internal Security Operations Command

Table 3 List of Defense Organizations ONLINE With JIT

(JIT's R&D Division, 1994)

B. JIT's ORGANIZATIONAL STRUCTURE

As envisioned by Supreme Command Headquarters, JIT's primary mission is to develop information technology tools, techniques and practices to integrate the information resources of all three Services. The goal is unifying command, control, communication and intelligence of the Royal Thai Armed Forces. To satisfy the mission requirement, JIT is organized into 6 divisions and 1 institution. The following are JIT's organizational units:

- **Administrative Division** is JIT's house-keeper. Personnel management, administrative activities and unit logistics operations are examples of the division's responsibilities.

- **Planning and Policy Division** develops and formulates strategic plans and policies for JIT. Introducing a computing resources standard into RTMoD's Services in one of the policies for initiating interoperability.

- **Software Development Division** deals with the software development life cycle of JIT's software systems. Requirements, design, coding, implementing and upkeeping software systems are the division's responsibilities.

- **Defense Information Technology Division** provides technical supports for C3I systems. Managing and developing necessary defense information technology systems is the division's mission. Office automation is an example.

- **Hardware Operations Division** is responsible for all hardware operations. Computer installation and maintenance are common tasks for the division.

- **R&D Division** performs computer R&D for JIT's future needs and acts as JIT's data communication expert.

- **Defense Computer Institute** provides RTMoD information technology knowledge. The institute organizes JIT's in-house training and provides external training support for RTA, RTN, and RTAF.

C. JIT'S COMPUTING RESOURCES

JIT has a computing capability greater than all of the Services combined. The explanation is purely based on the technical attributes of the computing resources. The computing resources belonging to the Services commonly involve personal computer systems and local area networks; these resources are limited in number. These computing resources are widely isolated and separated. Each organization in each Service develops its own information technology solutions. JIT is the only defense organization that centrally holds both manpower and computing resources. Table 4 illustrates JIT's computing capability, which is based on mainframe computer systems.

FEATURES	IBM 9121-320	IBM 4341-M02	CYBER 932-11
Main Memory	128 MB	8 MB	16MB
Processing Speed	22.5 MIPS	1.4 MIPS	8 MIPS
Operating System	MVS/ESA	VM/SP	NOS/VE
Secondary Storage	DSDA-20GB	DSDA-1GB	DSDA-828MB
Software	RDBMS-ORACLE	OA-PROFS	DBMS-IM/DM

Table 4 JIT's Computing Resources

(JIT's R&D Division, 1994)

The IBM 9121-320, which provides up to 24 data channels, offers the on-line information support for defense organizations throughout the country. It is the only computing resource that presently acts as database network hub for the RTMoD. More importantly, most of relational DataBase Management System software applications that are developed are based on this platform.

The Cyber 932-11 will provide similar services as the IBM 9121-320 after the technical configurated connections have been made. At the moment, the Cyber 932-11 is functioning as an intelligence information repository via DBMS IM/DM software. Because the DBMS software works superlatively well with text-format data, JIT also uses this computing resource for text retrieval.

The IBM 4341-M02 is used for office automation. At this stage of development, the system only provides internal services for JIT's senior executive officers (e.g., e-mail, electronic meeting, electronic scheduling and electronic data interchange). However, the system will be expanded to service all Supreme Command Headquarters's senior executive officers in the near future.

D. SOFTWARE ENGINEERING DEVELOPMENT IN JIT

Most software projects are managed by the Software Development Division. Developing office automation software is the exception. Automation software is the Defense Information Technology's responsibility. Few software engineering personnel participate in office automation software projects; the majority of software engineers are assigned to the Software Development Division.

To manage software engineering personnel, the Software Development Division organizes personnel into five groups, according to their skills and expertise: personnel, intelligence, operations & training, logistics and comptroller. This arrangement matches the organizational structure of the Supreme Command Headquarters (i.e., Joint Personnel, Joint Intelligence, Joint Operations & Training, Joint Logistics and Joint Comptroller).

Each software development group includes a software project manager, who has 10-to-15 years of experience in software development for a particular functionality, and 5 or 6 software engineers. In developing software products, each group follows the traditional "waterfall" model. Development starts with a user's requirement analysis. Most of this task is performed by the software project managers because of their long experience in the area. Sometimes data flow diagrams (DFD) may be employed. However, the software project manager dictates which software tools and practices to use and directs the development process. Junior software engineering personnel perform the time-consuming coding task. The Division relies heavily on the Oracle relational Database Management System as its primary software development tool. In other words, they are database developers. Cobal CICS, Assembler and Kick (Communications-ONLINE software application) are also used as development tools.

An interview suggested that there is no formal software project management. Software project managers rely on their prior experience in estimating project size and completion time. Evaluating project cost is not a common practice since there is no formal cost control management. It is very difficult for the Software Development Division to identify which costs are associated with what processes. In fact, it is rather peculiar for any defense organization to cost their products or services. The Software Development Division is no exception.

The management tool that controls a software project is the time-table; this more or less identifies the project milestones over its life cycle. This time-table can be altered at any time at the software project manager's discretion. Software quality assurance (SQA) is based on Oracle's debugging tools. No SQA procedure or technique is ascribed. Oftentimes, users report software defects. Furthermore, slippage is common because there is little software tracking and oversight or software configuration management.

IV. APPLICABILITY ANALYSIS OF SEI'S CMM FOR JIT

In this chapter, the applicability of SEI's CMM for the Joint Information Technology (JIT) Supreme Command Headquarters is investigated. The applicability model is developed as a conceptual framework. Applicability attributes are introduced to characterize the interaction between the CMM software technology and JIT. To capture JIT's organizational environment, the stakeholder principle is utilized. Finally, the interaction between applicability attributes and their features is analyzed using force-field theory.

A. METHODOLOGICAL FRAMEWORK FOR BENCHMARKING ANALYSIS

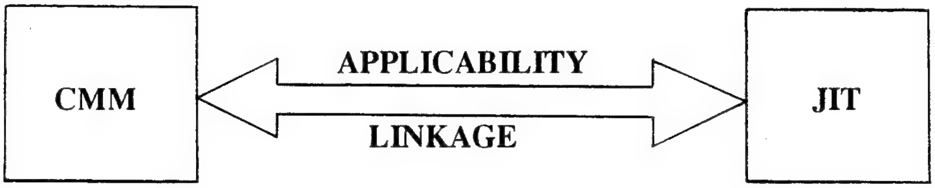

Figure 7 An Applicability Analysis Model

Applicability can be viewed as bi-directional linkage mechanism integrating the attributes of CMM and JIT. Figure 7 illustrates this operational definition. In order to fully understand the interactions between these two entities, applicability attributes are introduced and categorized into the three following groups.

- CMM Technological Applicability - characterizes the technical dimension of CMM in terms of its performance, technology shelf-life and the technical deficiencies of the model.

- CMM Economic Applicability - quantifies the outcomes of implementing the model in an organization. Investment costs and possible opportunity costs if the CMM has not been implemented are the prime features of this attribute.

- JIT Organizational Applicability - addresses specifically the readiness and capabilities of JIT for introducing CMM technology.

These three attributes are suitable for explaining the dynamism of interactions between CMM and its Thai counter-part, namely the JIT.

B. CMM TECHNOLOGICAL APPLICABILITY

In JIT's perspective, technological applicability can be divided up into three main features: performance, technology shelf-life and the model's deficiencies or shortcomings. These three basic features provide ample information concerning CMM's overall technological dimensions.

1. Performance

As mentioned in Chapter II, SEI's CMM-based assessment provides actions and guidelines for software process improvement (SPI) in an organization. To measure the outcomes of an SPI endeavor requires assessing and evaluating CMM. The performance feature is designed for this job. Considering JIT's organizational requirements, the performance feature is narrowed down to two important areas: 1) to what extent does

CMM improve software development cycle time, and 2) to what extent does CMM help an organization in detecting software defects.

A field study on the results of CMM-based software process improvement was conducted by the SEI's researchers in 1994 (Herbsleb et al, 1994). This research presents very rich and useful empirical evidence on the outcomes of adapting CMM in an organization. Of 13 organizations that participated in the study, 5 (Bull HN, Hughes Aircraft, Taxas Instrument, Schlumberger and Oklahoma City Air Logistics Center) agreed to publicly release the end results. Tables 5 and 6 illustrate the aggregate findings of the performance study.

Company	Number of Years in SPI	% Reduction in Development Time
A	3	15
B	6	23

Table 5 % Reduction Per Year in Calendar Time to Develop Software Systems (Herbsleb et al, 1994, p.12)

Although the companies are anonymous, it does not nullify the validity of CMM's overall performance outcomes. As Table 5 and 6 generally suggest, CMM generally improved the way the organizations produced their software products. Both the reduction

Company	Number of Years in SPI	% Reduction in Defects Reported
C	9	39
D	1.5	94
E	1	70
F	3.5	10
G	3.5	11

Table 6 % Reduction in the Number of Defects Reported by Customers Per SPI's Year (Herbsleb et al, 1994, p.13)

in development cycle time and reduction in defects, as reported by customers, are very impressive. For example, after 6 years of implementing CMM, Company B is capable of shortening its project's development life cycle by 23% of the total yearly calendar time. Company E reported that its software defects were reduced by 70% only one year after software process improvement was implemented.

As another example, Lockheeds's Missile and Space Company initiated CMM and compared its experience before and after implementation. In a typical size project of 500 thousand source lines of code (KSLOC), Lockheed experienced an average of 9 defects/KSLOC, which cost $32.5 million to correct. After 3 years of SPI, the company moved up from Initial Level (Level 1) to Defined Level (Level 3) and experienced only 1 defect/KSLOC. This cost only 6.5 million to correct (Springsteen, 1991, p.14). As a company progresses higher up along the maturity-level ladder, the technical performance feature increases substantially. This is a beneficial result of CMM.

2. Technology Shelf-life

Another crucial factor that needs to be considered before acquiring any software tools, techniques or practices is the expected useful life of that particular technology. To capture this, technology shelf-life is introduced. The determining factors that may affect shelf-life are: 1) the growing acceptance of the software process paradigm, 2) the endorsement of the CMM by US DoD, and 3) the spontaneity of the CMM in coping with changing software practices. By taking a closer look at these three areas, CMM's shelf-life can be estimated.

a. The Growing Acceptance of Software Process Paradigm

SEI's process assessment and improvement program have demonstrated that the core of CMM, which is the software process paradigm, underlines critical successes in dealing with a "software crisis." Its robustness and vigor yields a better way to develop software products. Hughes Aircraft, Westinghouse Electronic Systems, Raytheon

Equipment Division and NASA are among organizations that reap CMM's advantages (Herman and Lewis, 1993, p.12-16). It is worth reiterating that organizations which follow a systematic, repeatable, evolving process are more productive and produce products of higher quality than organizations whose processes are "ad hoc" or "chaotic" (Herman and Lewis, 1993, p.6-7). They "lack the explicit definitions that would ensure repeatability or allow systematic improvement" (Dowson, 1993, p.55).

Another indication of the wide spread use of CMM's software process paradigm is SEI's 1992 survey. This survey found that 75 percent of the 47 organizations which have undergone assessment and capability evaluation programs have been rated as Initial Level (Level I) (ad hoc and chaotic actions govern the development process). About 15 percent were Repeatable Level (Level II) and less than 10 percent have defined processes which fall within Defined Level (Level III). In other words, 90 percent of all sites studied are at Level I or II (Joyce, 1994, p.53;Baumert and Howard, 1993, p.102). This low average maturity agrees with Humphrey's assertion that "not enough attention is paid to the overall software development process itself" (Humphrey, 1992, p.28). He further stated that the ad hoc approach currently practice by most software companies "will not be sufficient to tackle the task of developing complex software systems for today and tomorrow" (Humphrey, 1992, p.28).

Interest in the software process paradigm has been growing at an accelerating rate; it is now the central topic for numerous groups and research projects. Evidence of its gaining popularity can be seen by the increase in international conferences and workshops on software processes. The notable examples are the semi-annual

International Conference on the Software Process and the European Workshops on Software Process Technology. In addition, several corporate R&D and Software Quality Assurance (SQA) groups have worked to standardize and improve the company's development process. Oftentimes, significant productivity and quality benefits are realized (Dowson, 1993, p.55-56).

b. The Endorsement of the CMM by the US DoD

The US DoD has played a pivotal role in pushing CMM technology since it studied the model in 1987 (Humphrey 1987). The original intent of the model was to assess the software capability of DoD contractors. In fact, DoD has accepted CMM as almost a de facto standard for assessing and evaluating its own software organizations and those of DoD contractors (Joyce, 1994, p.53).

Oklahoma City Air Logistics Center, Tinker Air Force Base represents a crown example of achievement from using CMM to improve its software process. Over one million dollars has been invested in SPI programs for the last 4 years. More than 4 million dollars has been saved as a result (Herbsleb et al, 1994, p.37-38). In fact, USAF policy requires all its software organizations to be at least Level III by the year 1998. This signifies that Air Force leadership is totally committed to SPI (Joyce, 1994, p.53). Moreover, SEI reported that 1482 people from 91 governmental organizations, 226 industry organizations and 21 academic institutions have participated in various software process assessment (SPA) activities (Humphrey and Curtis, 1991, p.45).

On the use of software capability evaluation (SCE), SEI reported that more than 200 people, representing 45 organizations, have been trained by SEI (Humphrey and Curtis, 1991, p.45). In a similar context, the US Department of Navy estimated that SCE has been used in "more than 20 acquisitions since late 1987, some of them involving contracts worth more than $100 million"(Rugg, 1993, p.36). In fact, SCE is becoming a common practice in the DoD acquisition community and its effects are spilling over to private industry. The rational is that a rating system that is good enough for the US government should also be good for industry (Bollinger and McGowan, 1991, p.26). Finally, both SCE and SPA will have a far greater impact in the coming years, especially on the DoD defense-related industry, because of DoD's minimum Level III software organization requirement for the source selection process (Abdel-Hamid, 1995).

c. The Spontaneity of the CMM's Transformation

SEI's unique ability to integrate and assimilate feedback generated from applying CMM in the software community enhances the integrity and usefulness of the model. This reflects the fact that "CMM is a living document" and is continuously measured and improved (Paulk et al, 1993, p.26).

The most significant recent transformation is merging SCE and SPA into CMM-Based Appraisal (CBA). This development will ensure the accuracy and consistency of the organizational software process appraisal results, which in the past have been inaccurate and inconsistent (Baumert, 1994, p.111). Furthermore, the People Management Capability Maturity Model (PM-CMM) has just been introduced to help software organizations with guidance on how to gain control of their processes for people management and human resource (HR) practices (Software Engineering Institute, 1995).

3. CMM's Deficiencies

While CMM has been used for almost a decade, technical shortcomings still exist in several areas. Some pertain to the model's architecture. Some arise using SPA and SCE. These model deficiencies can be summarized as follows.

- According to the software engineering academic community, CMM has not yet been validated and tested (Abdel-Hamid, 1995).[1] What seems to be SEI's approach to validating CMM is based on the knowledge acquired from extensive information it gathers from many process assessments and capability evaluations (Paulk et al, 1993, p.20). Such an approach does not constitute an acceptable scientific validation as accepted by the bono fide software engineering discipline (Abdel-Hamid, 1995).

- Several areas which are important to an organization's software engineering capability have been neglected in the CMM model. These areas are human resource management (e.g., selecting, hiring, developing and retaining competent personnel), physical working setting (e.g., lighting, work place layout etc.), software engineering methods and tools (e.g., requirement, design, support, and application development tools), and product and technology constraints (e.g., hardware experience, language proficiency, reuse and maintenance experiences) (Paulk et al, 1993, p.22;Bollinger and McGowan, 1991, p.30;Springsteen, 1992, p.20). In response to the deficiencies in human resources areas, SEI launched its PM-CMM workshop in 1994, but there is no empirical data to support the outcome of this initiative at this time (Software Engineering Institute, 1995).

[1] Personal interview with Tarek Abdel-Hamid, Professor of Software Engineering Management, Naval Postgraduate School, Monterey, CA, 23 March 1995.

- There appears to be limited uniformity between SCEs as practiced by US Army, Air Force and Navy acquisition organizations (Baumert, 1991, p.79). To correct this shortcoming, SCE and SPA are combined into CMM-Based Appraisal (CBA) with a Common Rating Framework (CRF) as a diagnostic tool (Software Engineering Institute, 1995). Like PM-CMM, the results of this new development are still unknown.

- CMM is effective in improving software processes for large software organizations (e.g., Bell Labs, Lockheeds, AT&T, IBM, McDonnel Douglas, Northrop and the commercial divisions of several aerospace companies) but not for small software organizations with less than 50 software practitioners (Springsteen, 1992, p.10-12). Ineffectiveness derives from the lack of resources required to implement many key process areas (KPAs) in the CMM model in order to move to the next higher level (Broadman and Johnson, 1994, p.331-340). For that reason, CMM places these small software organizations as Initial Level (Level I); they failed to meet the criterion of Repeatable Level (Level II) (Bollinger and McGowan, 1991, p.30).

C. CMM ECONOMIC APPLICABILITY

Both CMM's implementation cost and return on investment are regarded as the most important financial factors in determining the model's economic applicability. Appraisals of CMM's economic applicability draw heavily from the SEI's Empirical Methods Project working group report (Herbsleb et al, 1994). All information pertaining to individual organizations is strictly confidential and only released with the permission of that organization. As one would expect, the literature only reports successful implementations of CMM, particularly in terms of economic payoffs. This is not to say that the CMM always results in an economic success, only that there are no published examples to contrary.

Company	**Number of Years in SPI**	**Yearly Investment (Thousands of Dollars)**
P	6	1,203
Q	2	245
R	6	155
S	3.5	49
T	2	516

Table 7 Thousands of Dollars per Year Spent on SPI

(Herbsleb et al, 1994, p.9)

Table 7 suggests that the range of the CMM's yearly implementation cost is between $49,000 and $1,203,000. On the other hand, an average yearly investment in SPI is approximately $433,000.

Company	Numbers of Years in SPI	Return of Investment in Dollar Value
U	3.5	4.0
V	6	5.0
W	6	4.2
X	5	6.4
Y	3.5	8.8

Table 8 Return of Investment of SPI Efforts

(Herbsleb et al, 1994, p.14)

The return on investment is derived from the cost avoidance or cost saving, including rework and duplication of function (Herbsleb et al, 1994, p.38). These five companies reported that CMM improved their capability to detect defects, especially during the early phase of the development life cycle. This generates cost savings or a return on investment. Table 8 implies an average of a $5.68 return on every dollar invested. This represents a substantial savings from a software organization's perspective.

D. JIT ORGANIZATIONAL APPLICABILITY

JIT's organizational environment can be adequately assessed by performing a simple stakeholder audit (Roberts 1995).[2] A stakeholder is defined as "any group or individual who can affect or is affected by an organization and its policies" (Mitroff, 1983, p.22). The implication is that the key to successfully adopting CMM into JIT depends on the level of satisfaction and support by key JIT's stakeholders. In their article, The Stakeholder Audit Goes Public, published in 1989, Nancy C. Roberts and Paula J. King conceptualize a stake as "based on the idea of one's having something to lose or gain in a given situation, and therefore the nature of the stake depends on the issue at hand" (Roberts and King, 1989, p.66). In simple terms, a stake is the stakeholder's claim on the organization. Stakes can be either tangible (money, material resources) or intangible (time, prestige, self-esteem) or both (Roberts and King, 1989, p.66).

To uncover the stakeholdrs in JIT, a focal organizational approach was applied. This approach seeks to identify the individuals and organizations who have an important relationship with the focal organization (Mitroff, 1983, p.35). Three groups who directly influence or have stake in adopting CMM are: 1) JIT senior executive officers (SEOs), 2) Software Development Division personnel and 3) clients or users of the software developed by JIT. Specifically, SEOs and software engineering personnel of the Software Development Division are internal stakeholders; clients are external stakeholders. All

[2] Personal interview with Nancy C. Roberts, Professor of Organizations and Management, Naval Postgraduate School, Monterey, CA, 15 April 1995.

three alter JIT's setting and "are elements of the direct-action environment"(Stoner and Freeman, 1992, p.62). Figure 8 outlines JIT's environment and shows the influence of direct-action elements.

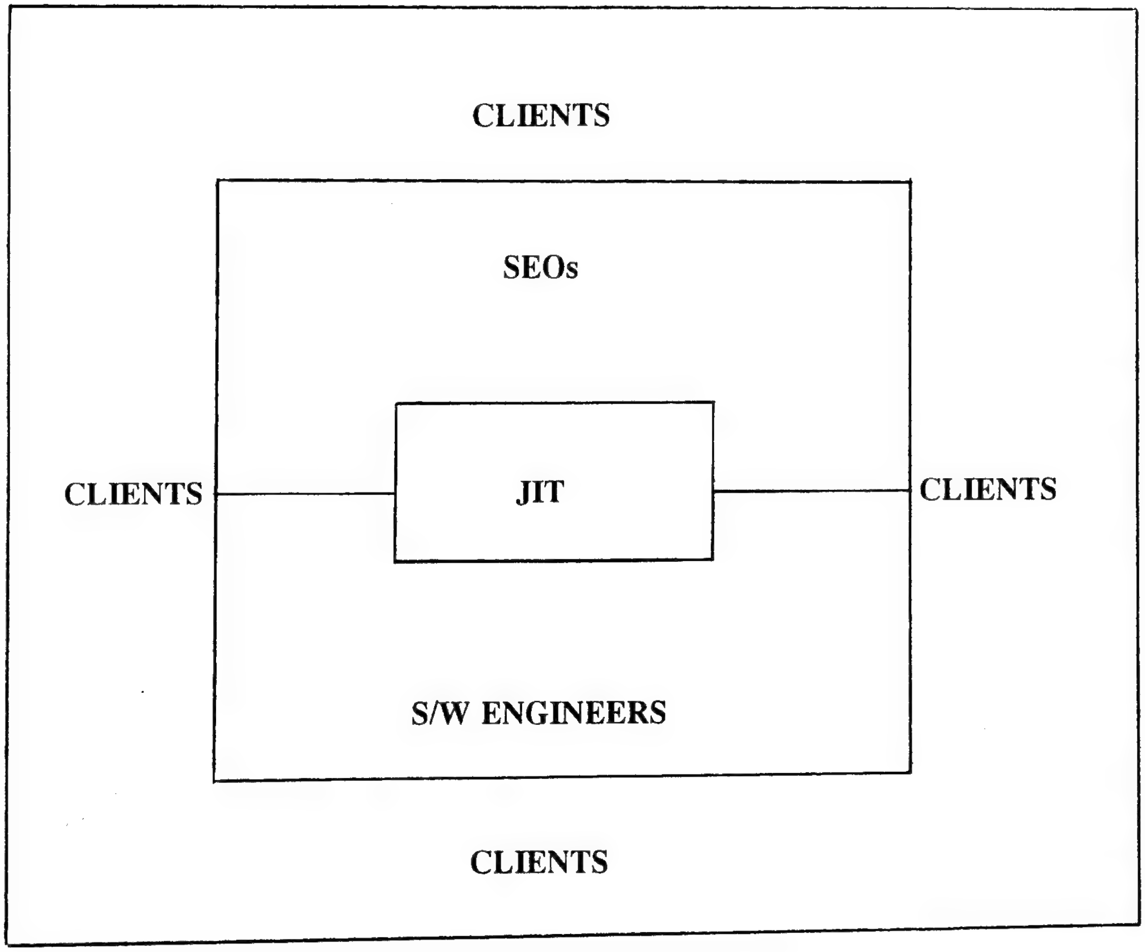

Figure 8 The Direct - Action of JIT's Stakeholders

(Adapted and Modified from Stoner and Freeman, 1992, p.62)

1. JIT's SEOs

SEOs command and manage overall JIT resources. They have the highest stake if CMM is adopted. Bureaucratic and centralized top-down management is a common practice in Royal Thai military organizations. JIT is by no means an exception. Studying their purposes and motivations can help explain the likelihood that CMM will be adopted. As discussed previously, the lessons learned show that SPI is a long-term effort which "requires leadership and long-term commitment from executive management"(Herbsleb et al, 1994, p.20).

2. Software Development Division Personnel

As mentioned earlier, software engineering personnel are internal stakeholders. The principle resources that they command are their skills and special knowledge in developing software products. Due to the high discrepancy in rewards and compensation between the private and public sector, there is less incentive for software practitioners to pursue their careers in public sector. During the last 5 years, JIT suffered a personnel brain-drain; the turnover exceeded 30%. JIT found itself in the difficult position of luring new computer graduates into the organization. To reverse the trend, JIT collaborated with academic institutions like Ramkamgheng University and King Mongkutklao Institute of Technology to provide educational support for its personnel; JIT also institutionalized in-house software engineering training. There is no definitive data to support the effectiveness of such policy at the moment. One imperative conclusion in this situation is that personnel shortages make a command seriously consider the stakes of these software engineering personnel.

3. Clients or Users

Clients are the least understood stakeholder in JIT's setting. The primary mission of JIT is to standardize the information technology resources of all Services (Army, Navy and Air Force) so that the unified command, control, communication and intelligence can interact. To a certain extent, its mission is servicing its clients. All software has been developed configured to that end. A good example is the development of the Defense Military Information System (DMIS). In this project, JIT developed tiers of Oracle's software applications to integrate a pool of separate data files. DMIS provides the information needed by all Royal Thai Military organizations, particularly Ministry of Defense (MoD) SEOs and their General Staffs(GSs). Those who use the common software product which was developed with the unified or integrated approach are JIT's clients.

E. APPLICABILITY ATTRIBUTES IN OPERATION

Force field analysis provides a framework for assessing the interactions between applicability attributes and their features (Thomas, 1995).[3] Generally, force field analysis is "a tool for analyzing organization attitude/cultural readiness to accept change (American

[3] Personal interview with Ken Thomas, Professor of Organizations and Management, Naval Postgraduate School, Monterey, CA, 10 April 1995.

Productivity & Quality Center, 1995, p.6-10). Introducing CMM into JIT provides a meaningful test-bed case for utilizing this tool.

According to Kurt Lewin's force field theory, every situation or behavior is the result of a balance of conflicting forces: drivers and restrainers (Stoner and Freeman, 1992, p.410). The drivers push one way, the restrainers push the other. The performance that emerges is a reconciliation of two sets of forces. Importantly, an increase in the driving forces might increase performance, but it might also increase the restraining forces. Therefore, the tension or degree of conflict in an organization is likely to increase (Huse, 1975, p.48). Figure 9 shows the dynamic impact of introducing CMM in JIT, based on the force field theory. The arrows represent the vectors, or forces, applied to JIT's organizational state of equilibrium. The length of the arrows indicates the strength of the forces.

1. Forces for Change

There are three major driving forces influencing JIT's state of equilibrium: 1) CMM's technical strengths and applicability, 2) return on investment from implementing CMM-based software process improvements (SPI), and 3) JIT's client software products and service needs. The first two driving forces come from CMM technology itself, the third stems from JIT's organizational environment as defined in the stakeholder analysis. These three forces of change will be discussed below.

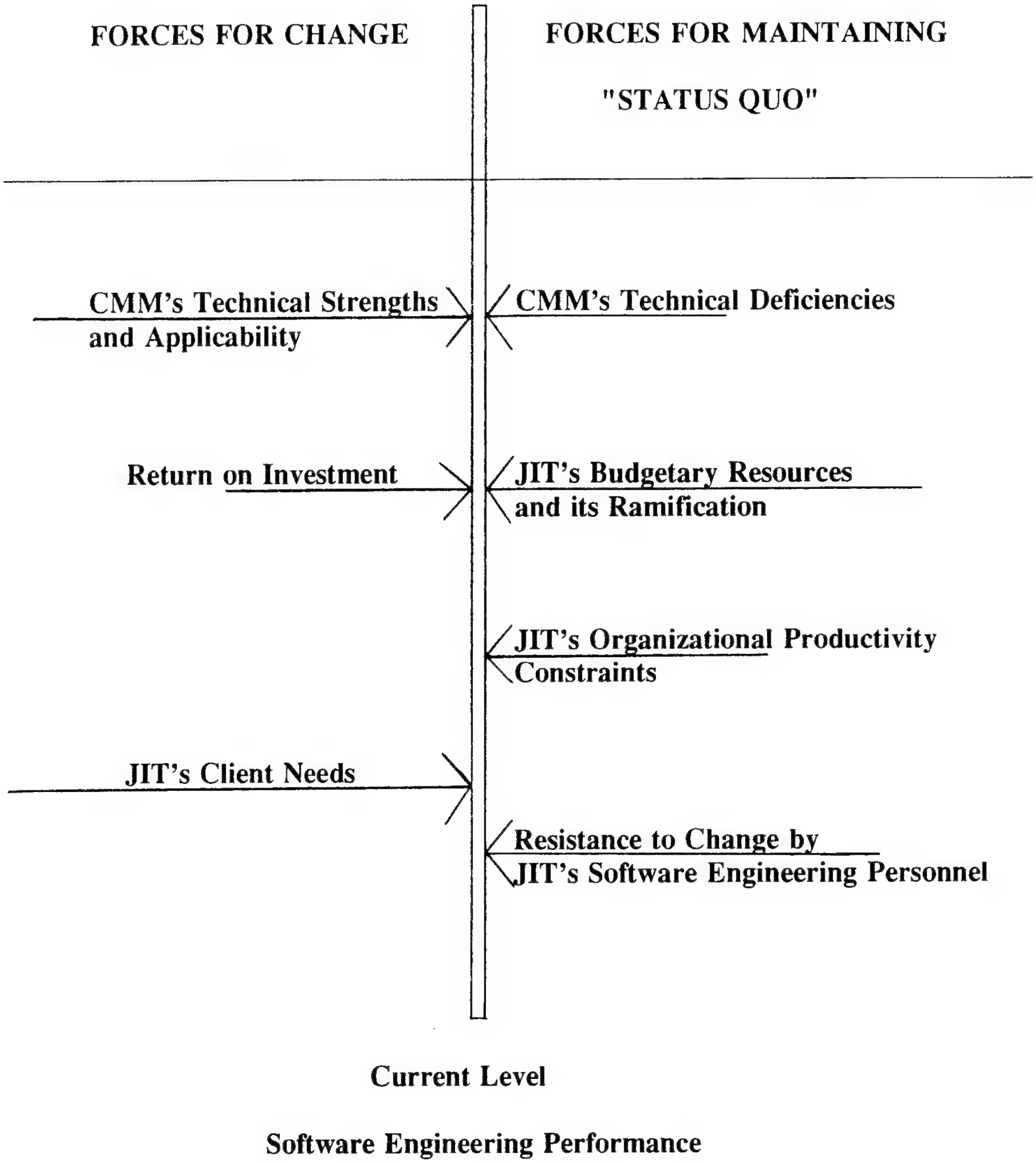
FORCES FOR CHANGE
FORCES FOR MAINTAINING
"STATUS QUO"
CMM's Technical Strengths
and Applicability
CMM's Technical Deficiencies
Return on Investment
JIT's Budgetary Resources
and its Ramification
JIT's Organizational Productivity
Constraints
JIT's Client Needs
Resistance to Change by
JIT's Software Engineering Personnel
Current Level
Software Engineering Performance

Figure 9 Force Field Diagram for the Introduction of CMM in JIT

(Adapted and Modified from Huse, 1975, p.50;Stoner and Freeman, 1992, p.411)

a. CMM's Technical Strengths and Applicability

CMM has demonstrated its technical superiority in two areas that JIT can exploit to improve its software production performance. Specifically, CMM's strengths are its effectiveness in shortening the software development life cycle and reducing software defects. Field experience in implementing CMM's key process areas (KPAs) showed that the software development life cycle calendar time has been shortened by at least 20% on a one-year software project. In other words, JIT would develop software application products 52 days faster than with its existing process. This would alleviate the software backlog that JIT is now experiencing. Similarly, CMM-based SPI has helped organizations improve their capability to detect software defects, especially during the early stage of the software development life cycle. SEI's research showed that the number of software defects were reduced by 12% on average (Herbsleb et al, 1994, p.13). JIT would reap this benefit by not having to rework its software application systems. There will be fewer complaints from clients on the quality of JIT's software products and services. Importantly, JIT could utilize its limited resources on other important matters.

CMM is based on a continuous process improvement approach. Deming's statistical process control principles, Shewhart's Plan-Act-Check-Do process improvement cycle, Crosby's Quality College Model and Juran's Quality Trilogy are CMM's basic building blocks. In simple words, quality management philosophy is the driving force underlining CMM's development. As a result, US DoD has accepted CMM as a de facto standard for assessing and evaluating its own software organizations and those of DoD contractors (Joyce, 1994, p.53). Moreover, quality management has been successfully practiced by many branches of both US federal and States governments. On the federal level, this includes the Department of Defense, Department of Commerce, Department of Labor and the Internal Revenue Service. Successful state government quality efforts include are Arkansas, Minnesota and New York (Hunt, 1993, p.112-180). This management paradigm recognizes the richness of human potential. Employee empowerment and teamwork, customer focus, top management leadership and support, and quality assurance are among the fundamental concepts in quality management. CMM is designed to encompass these quality criteria.

Presently, JIT's clients are not satisfied with the quality of software database services and software application products. Some are starting to seek their own information technology solutions. This suggests that JIT's bureaucratic top-down management can not adequately cope with key software engineering problems. If this trend continues, it will jeopardize JIT's organizational objectives and commitments. Full budgetary support might not be offered in the future. JIT's existence might be threatened. In addition, JIT has difficulty retaining its valuable software engineering personnel. One explanation might be that JIT's hierarchical management structure deemphasizes its employees. As mention earlier, software engineering personnel usually code mundane database software applications. They perceive their software project managers as bosses who direct and control all software development activities. Employee involvement in the software development life cycle has been neglected. More importantly, JIT's bureaucratic software development management undermines the intellectual horsepower of software engineering personnel; these personnel would be created and enriched under CMM's philosophy.

b. Return on Investment

Field experience from implementing CMM-based SPI showed a substantial return on investment is generated through cost avoidance or cost saving, including less rework and duplication of functions (Herbsleb et al, 1994, p.38). The average estimated return is $5.68 return for every dollar invested. Although the return on investment ratio is very impressive for the private US software industry, it is not as impressive from JIT's point of view. JIT rarely practices formal cost management. JIT is basically a driven political-bureaucratic organization. As one of the SEOs pointed out," The SPA would pay off for the big US software corporations but not with little organization like ours where the budget is always short and we have many tasks to perform."[4] This undefined and peculiar situation diminishes the importance of return on investment as a driving force.

c. JIT's Client Needs

Client needs is a significant driving force for change. Clients derive their utility from the software products and services provided by JIT's Software Development Division. This reciprocal relationship creates a change tension. In fact, clients are the main driving thrust encouraging JIT to develop and deliver quality software products and database services.

[4] Telephone interview with an anonymous officer, Joint Information Technology, Supreme Command Headquarters, Bangkok, Thailand, 18 April 1995. Anonymity has been requested.

Currently, JIT is experiencing the difficulties in delivering its products and services on schedule. The software backlog is about two-to-four years. Some of the software has not met the client's needs and requirements. Oftentimes, software engineering personnel were asked to maintain the software systems due to the software defects. This situation ties up most of JIT's Software Development Division's limited resources. More importantly, some clients are developing independent channels to satisfy their information technology needs.

Database service is another area where JIT's clients are demanding improvement. There has not been much progress in this area. In fact, there has been an outcry of complaints. Some clients were not able to retrieve the information they needed. Some clients' computer systems could not link with JIT's main database system. These unsatisfied clients, especially the top level Ministry of Defense SEOs, will have a big impact on JIT's organizational change.

2. Forces for Maintaining the "Status Quo"

There are four restraining forces upholding JIT's organizational "quasi-stationary equilibrium" (Huse, 1975, p.50) : 1) CMM's technical deficiencies as envisioned by JIT's internal stakeholders, 2) JIT's budgetary resources, 3) JIT's organizational productivity constraints, and 4) resistance to change by JIT's software engineering personnel. The first restraining force involves the technical shortfalls of CMM technology as seen by the SEOs and software engineering personnel. The last three originate within JIT's organizational environment.

a. CMM's Technical Deficiencies

As viewed technically by the JIT's internal stakeholders (e.g., SEOs and software engineering personnel), CMM in not without technical flaws. The model deficiencies are used to counter its productive effectiveness. JIT's current software engineering environment uses DBMS's software application tools intensively, but also faces difficulty in recruiting and retaining personnel. Moreover, it consists of less than 40 software practitioners. These are the technical areas where CMM is not particularly serviceable. As mentioned earlier, the People Management Capability Maturity Model (PM-CMM) National Workshop has only just been introduced. There has been no field experience to support this new development. There is no real technical need or motivation to adopt the CMM technology immediately. According to one of the SEOs, "The SPA would work fairly well for the giant US software corporations but not with our little organization that has too few software professionals and, more importantly, the CMM is still evolving and immature in some areas. It is highly risky for JIT to invest in now" (Anonymous officer, 1995). It is plausible that the SEOs will adopt a "wait and see" or "let the dust settle" strategy in this circumstance.

b. JIT's Organizational Productivity Constraints

With respect to the hypothetical situation that CMM is adopted, this will put JIT on the low end of another learning curve. Some productive capability may have to be sacrificed before JIT is able to move along the new learning curve. There are no guarantees as to how fast a person, group or organization can learn (Przybylinski and

Fowler, 1987, p.132-133). Moreover, the lessons learned pointed out that it may take 10 years or more to build the foundation and a culture to continuously improve the software process (Herbsleb et al, 1994, p.33). Faced with difficulties in measuring productivity and the rapid advances in software technology, it is reasonable for the SEOs to resist CMM. Conservatively speaking, they would rather deal with old production technology with which they are comfortable.

Implementing CMM is bound to be a political battle for the Software Development Division because its personnel lack the required skills, particularly for establishing CMM's KPAs (key practice areas). As one software engineer commented,"It will be easy to rate JIT as Level I, and it is, so how do we go about to improve when the resource is so limited and so few people, may be 1 or 2, would be qualified to do the SPA job, not to mention about the associated training cost. It is an uphill task for an immature organization like ours" (Pungboon, 1995).[5]

c. *Resistance to Change by JIT's Software Engineering Personnel*

It is also natural for software engineering personnel to resist the change. Excuses include,"CMM looks to complicated and it will not be workable here," and "the model does not include the tools that we dearly need in our projects"(Pungboon, 1995). The "not invented here" syndrome is also quite pervasive in JIT. In the past, software engineering personnel made strenuous efforts to overcome internal problems and gained recognition for their efforts. It will be difficult for them to give up their investment in the

[5] Telephone interview with Captain Prapass Pungpoon, JIT's Software Development Division, Supreme Command Headquarters, Bangkok, Thailand, 10 April 1995.

solution that they discovered, considering their efforts and the recognition they received for their ingenious use of existing resources.

Practically speaking, CMM is a radical departure from their current practice. The technical-cultural barrier poses a hindrance as "people come to value and rely upon the systems which were already embedded within their established work and computing arrangement"(Przybylinski and Fowler, 1987, p.132). Keeping their computing infrastructure and their beloved DBMS tools may become important, adopting CMM would naturally be a secondary priority (Przybylinski and Fowler, 1987, p.132). CMM technology does not represent a solution that they currently need badly (Pungboon, 1995). As long as adopting CMM does not reenforce the established software engineering personnel's work routine, it is very likely that the CMM will not be adopted. The confirmed truth is that "adopting a new tool or techniques will likely be self-serving and constrained by circumstances within their work place, rather than according to some superlative technical characteristics or intrinsic value ascribed to the technology" (Przybylinski and Fowler, 1987, p.131).

d. JIT's Budget Resources and its Ramifications

One draw back to the CMM technology, as seen by SEOs, is the high initial cost. SPA costs Hughs Aircraft company at least $45,000 (Herman and Lewis, 1993, p.12). From JIT's budgetary point of view, it would be difficult to devote these funds, not to mention the follow-up software process investment. Considering the US Air Force Oklahoma City Air Logistics Center, it costs Tinker OC-ALC more than $1 million for the SPI activities. That is equivalent to half of JIT's annual budget. Moreover, calculating the return on investment on the avoided cost of rework on defects is not totally applicable for JIT's software engineering. Removing JIT's database software defects is less costly and critical than reworking the complex real-time software which is being developed by companies like Boeing or Northrop.

CMM technology is a new software development philosophy. This tends to be adopted more slowly than more visible innovations such as hardware or software based innovations (Bayer and Melone, 1987, p.17). The mentality of the SEOs is measured by immediate or visible outcomes. "We would rather spend that amount of money on hardware, software or other office automation equipment, where the results are so visible that anyone, especially MoD SEOs, can see how JIT achieves its mission," one of SEOs commented (Anonymous officer, 1995). The benefits from adopting CMM are far less important than the alternative investment SEOs would select.

Software engineering personnel will react like the SEOs, "We can not foresee the beneficial outcome by investing in the CMM, but we do know with certainty that if that money were alternatively spent on what we really need (e.g., CASE tools, 4GL and graphic cards), we can accomplish faster and greater" (Pungboon, 1995). Software engineering personnel would gain more in the near term if CMM is not adopted; the foregone money could be invested in items for which they have pressed hard for a long time.

3. Conclusion of Applicability Analysis

Since the algebraic sum of the restraining forces (i.e., CMM's technical deficiencies, JIT's budgetary resources and its ramifications, JIT's organizational productivity constraints, and resistance to change by JIT's software engineering personnel) is greater than the driving forces (i.e., CMM's technical strengths and applicability, CMM's return on investment, and JIT's client needs), the force field theory suggests that no change is likely (Huse, 1975, p.48). In simple terms, JIT's internal stakeholders (SEOs and software engineering personnel) would like to maintain their status quo. CMM would not provide them any benefits but rather tie up their existing resources. This does not mean that CMM technology is not useful to JIT. It merely suggests that adopting CMM successfully means the stakeholders' objectives and motivations needed to be met. CMM must also demonstrate the ability to correct JIT's perceived deficiencies: human resource management, software application tools and techniques and the size of software organization.

F. SUMMARY

The applicability model was introduced as a conceptual analytical framework. Its precept is based on the simple linkage between CMM technology and JIT's organizational environment. To be more specific, CMM technological and economic applicability attributes were examined relative to JIT's perspective. JIT's organizational boundary was assessed under the stakeholder concept. Three groups of JIT's stakeholders were identified: SEOs, software engineering personnel and clients. These are the groups of people who have direct influence over introducing CMM into their organization. To illustrate the interactions between the CMM technological and economic applicability and JIT's organization applicability (i.e., JIT's stakeholders), force field theory was used as a diagnostic tool.

The results of the force field analysis suggests that the strength of the resisting forces (i.e., CMM's technical deficiencies, JIT's budgetary resources and its ramifications, JIT's organizational productivity constraints, and resistance to change by JIT's software engineering personnel) is greater than the driving forces (i.e., JIT's client needs, CMM's return on investment, and CMM's technical strengths and applicability). This means that change in JIT is unlikely. SEOs and software engineering personnel would probably work to maintain the status quo. They would gain fewer short run benefits if CMM is adopted. However, if the clients, particularly the Ministry of Defense SEOs who command both positional and material resources, recognize the future strategic importance of CMM technology, and provide total commitment in terms of budgetary resources, visibility and leadership, then CMM may be adopted. This is more likely if CMM can demonstrate successful empirical outcomes to counterbalance the deficient areas about which JIT is most concerned, namely, human resource management, software application tools and techniques and the CMM's effectiveness in a small organization. If these conditions are not met, it is unlikely that the resisting forces can be reduced. Consequently, change would be ill advised.

V. SUMMARY AND RECOMMENDATIONS

The previous chapter analyzed the interactions between CMM's capabilities and JIT's organizational stakeholders. This chapter uses the results of this analysis to answer the research questions. The secondary research questions are addressed first, followed by the primary question. Recommendations will be presented. Finally, areas for further research are identified.

A. ANSWERS TO THE SECONDARY RESEARCH QUESTIONS

The secondary research questions were answered in previous discussions. They are summarized in the following paragraphs.

1. What is the Capability Maturity Model?

As outlined in Chapter II, CMM is a framework which describes the organizational characteristics of a software development process and provides action plans to help an organization evolve and improve. To enhance organizational software development and capability, key process areas (KPAs) must be institutionalized according to their maturity level. By focusing on targeted KPAs, an organization can steadily improve its process capability and move up to the higher maturity levels. In short, CMM is a tool to assist software managers in their software development process and facilitate continuous process improvement.

2. What is JIT's Current Software Engineering Technology?

Chapter III described that JIT's engineering personnel predominantly use the Database Management System (DBMS) and its software application tools. This reflects the nature of the client's system requirements. Software project planning is in its infancy. There is no formal software project tracking or oversight. The only metric used to manage a software project is the schedule or time-table; this is simply the estimated time to complete each stage of the software development life cycle. In addition, there is no standard procedure for software quality assurance (SQA). The users normally report defects observed during operational use.

3. What Criteria Should be used to Assess CMM's Applicability Within JIT's Organizational Setting?

Chapter IV developed two important attributes to assess the CMM technology: technological and economic applicability. Technological applicability deals with CMM's performance, the expected useful life of the technology and the model's shortcomings. Economic applicability represents the cost and return in investment from implementing software process improvement, expressed in dollar terms.

4. What Approaches Should be Used in Analyzing JIT's Organizational Attitude/Cultural Readiness for Introducing CMM?

Chapter IV suggested a stakeholder audit as the basis for evaluating JIT's organizational attitude/cultural readiness for introducing CMM. SEOs, software engineering personnel and clients are the stakeholders having direct influence over this proposed change. These three stakeholders interact over the technology transfer policy as described in the force field analysis.

B. THE PRIMARY RESEARCH QUESTION

The primary research question is: "Is CMM applicable for JIT?" The analysis in Chapter IV suggests that CMM's value to JIT is dubious. Two principle stakeholders, SEOs and software engineering personnel, prefer to maintain the status quo. They do not recognize much gain from introducing CMM. JIT does not appear primed to acquire and adopt CMM. Justification for this conclusion can be summarized as follows:

- CMM technology does not exhibit strengths in the areas with which JIT is most concerned. Human resource management is excluded from the model. The model does not take software application tools, software techniques and personnel computing experience into consideration when assessing and evaluating the software development process. Lastly, operating CMM in a small software organization, which employs fewer than 50 software professionals, is outside the current experience base.

- Since CMM is a new software development philosophy, it is adopted more slowly than many visible hardware or software innovations. This contradicts the SEO's management practice. Immediate and visible results are emphasized.

- CMM is based on a different world view. Software process quality, continuous process improvement, teamwork and empowerment are alien concepts to JIT's organizational culture and top-down bureaucratic management environment. Adopting CMM may create tension and conflict in the organization.

- CMM technology does not represent an immediate solution for the software engineering personnel's problems. Moreover, it does not reenforce their established work practices. Adopting CMM may compound the existing problems. Undesirable repercussions may result.

C. RECOMMENDATIONS

According to force field theory, the best strategy for promoting proposed change in an organization is not to increase the driving forces. This may propagate more internal restraining forces. Rather, the strategy is to decrease the strength of the existing restraining vectors. SEOs and software engineering personnel must be encouraged to fully recognize CMM's merits. To accomplish this, these recommendations are offered:

1. Utilize educational technology as an agent for change. JIT's technology gatekeeper should establish formal procedures for disseminating information about the software process paradigm. JIT's Defense Computer Institute might provide information about CMM technology. Short courses about KPAs should be included in in-house training program.

2. The expected useful life of CMM technology is supported by the US Department of Defense's endorsement. It is a promising technology since its validity is based on the practical experience of the software engineering community. For that reason, JIT's technology gatekeeper should establish an arms-length relationship with SEI, particularly for the People Management Capability Maturity Model project (PM-CMM).

3. Software process improvement manifests itself in improving productivity. Technology gatekeepers might consider developing JIT's own SPA for assessing its internal software development. After KPAs have been identified, one or two can be implemented on a trial basis. The trial-ability will lower the risk of adoption.

4. As mention before, successfully implementing CMM requires cultural change. To achieve the low end of the continuum, a PDCA cycle (Plan, Do, Check, Act) should be strongly encourage, especially with software engineering personnel. This practice will provide a basic building block for continuous quality improvement (CQI). This is an integral part of CMM.

D. AREAS OF FURTHER RESEARCH

The limited scope of this thesis precluded further study of several issues. Several promising areas for continued research and study are:

- Is there any additional way to assess JIT's organizational attitude/cultural readiness to accept CMM technology?
- Is there any significant difference between SEI's software development process appraisal and a tailored one?
- Does the SPI implementation cost vary with the size of software organization? Is there any conceivable way to reduce the cost?
- What additional management approaches should be considered in overcoming the cultural and technical barriers?
- Are there other formal methods or tools for assessing CMM technology?
- What are the impacts of the PM-CMM for JIT? Does this technology represent a viable solution for problems facing JIT?

LIST OF REFERENCES

Abdel-Hamid, Tarek and Stuart E. Madnick, **Software Project Dynamics: An Integrated Approach**, Prentice Hall, Englewood Cliff, NJ, 1991.

American Productivity & Quality Center, **Tools and Techniques for Effective Benchmarking Studies**, 1995.

Bayer, Judy B. and Nancy Melone, **Adoption of Software Engineering Innovations in Organizations**, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, April 1989.

Bamford, Robert C. and William J. Deibler II, "Comparing, Contrasting ISO 9001 and the SEI Capability Maturity Model," **Computer**, October 1993.

Baumert, John, "New SEI Maturity Model Targets Key Practices," **IEEE Software**, November 1991.

Bollinger, Terry B. and McGowan Clement, "A Critical Look at Software Capability Evaluations," **IEEE Software**, July 1991.

Broadman, J. G. and D. L. Johnson, "What Small Business & Small Organization Say about the CMM," **ICSE-16, 16th International Conference on Software Engineering**, IEEE Computer Society Press, May 1994.

Dickeroff, William G. and William J. Sommers, **The Adaptation of the SEI's Capability Maturity Model to the Air Force Software Acquisition Management Process**, Air Force Institute of Technology(AU), Wright-Patterson AFB, OH, 1992.

DoD Joint Service Task Force on the Software Initiative, **A Candidate Strategy for the SEI**, Washington D.C., March 1983.

Dowson, Mark, "Software Process Themes and Issues," **Proceeding of the Second International Conference on the Software Process**, IEEE Computer Society Press, February 1993.

Herbsleb, James, et al, **Benefits of CMM-Based Software Process Improvement: Initial Results**, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, August 1994.

Herman, Brain G. and Raymond J. Lewis, **An Evaluation of the Software Engineering Institute's Capability Maturity Model As a Framework for Software Process Improvement At Three Air Logistics Centers**, Master's thesis, Air Force Institute of Technology(AU), Wright-Patterson AFB, OH, 1993.

Humphrey, Watts S. **Introduction to Software Process Improvement**, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, June 1992.

Humphrey, Watts S. and Bill Curtis, "Comments on 'A Critical Look'," **IEEE Software**, July 1991.

Humphrey, Watts S. and W. L. Sweet, **A Method for Assessing the Software Engineering Capability of Contracts**, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, September 1987.

Hunt, Daniel V., **Quality Management for Government: A Guide to Federal, State, and Local Implementation**, ASQC Quality Press, Milwaukee, WN, 1993.

Huse, Edgar F., **Organization Development and Change**, West Publishing Co., New York, 1975.

Joint Information Technology's R&D Division, **JIT's Organization and its Missions**, JIT, Supreme Command Headquarters, Bangkok, Thailand, 1994.

Joyce, Endoso, "AF Center Makes Software Development Grade," **Government Computer News**, 12 December 1994.

_______, "Army's Fort Lee Center Earns High Rating for Software Development," **Government Computer News**, 30 May 1994.

Kitfield, James, "Is Software DoD's Achilles's Heel?," **Military Forum**, July 1989.

Mitroff, Ian I., **Stakeholders of the Organizational Mind: Toward a New View of Organizational Policy Making**, Jossey-Bass Inc., San Francisco, CA, 1983.

Paulk, Mark C., et al, **Capability Maturity Model for Software, Version 1.1**, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, 1993.

_______, Capability Maturity Model, Version 1.1, **IEEE Software**, July 1993.

_______, **Key Practices of the Capability Maturity Model, Version 1.1**, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, 1993.

Przybylinski, Stan and Priscilla J. Fowler, eds, **Transferring Software Engineering Tool Technology**, IEEE Computer Society Press, November 1987.

Raguindin, Ferdinand M., **Selecting A Software Capability Evaluation for Weapons Acquisition**, Master's thesis, Naval Postgraduate School, Monterey, CA, September 1994.

Roberts, Nancy C. and Paula J. King, "The Stakeholder Audit Goes Public," **Organizational Dynamic**, Winter, 1989.

Rugg, David, "Using Capability Evaluation to Select A Contractor," **IEEE Software**, July 1993.

Sims, David, "Motorola India Self-Assesses at Level 5," **IEEE Software**, March 1994.

Software Engineering Institute, **About the SEI**, [Online], Available http://www.sei.cmu.edu/Front Door/html, 1994.

Springsteen, Beth, **Policy Assessment for the Software Process Maturity Model**, Institute for Defense Analysis, Alexandria, VA, August 1992.

_______, **Software Maturity Applied to SDI**, Institute for Defense Analysis, Alexandria, VA, September 1991.

Stoner, James A.F. and R. Edward Freeman, **Management**, Prentice Hall, Englewood Cliffs, NJ, 1992.

Werth, Laurie H., **Lectures Notes on Software Process Improvement**, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, February 1993.

_______, "Introduction to Software Process Improvement," **Lectures Notes on Software Process Improvement**, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, February 1993.

Zubrow, David, et al, **Maturity Questionnaire**, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, 1993.

Interview between Tarek Abdel-Hamid, Professor of Software Engineering Management, Department of Systems Management, Naval Postgraduate School, Monterey, CA and author, 23 March 1995.

Interview between Nancy C. Roberts, Professor of Organizations and Management, Department of Systems Management, Naval Postgraduate School, Monterey, CA and author, 11 April 1995.

Interview between Ken Thomas, Professor of Organizations and Management, Department of Systems Management, Naval Postgraduate School, Monterey, CA and author, 4 April 1995.

Telephone Interview between Anonymous officer, JIT's Plan and Policy Division, Supreme Command Headquarters, Bangkok, Thailand and author, 18 April 1995.

Telephone Interview between Prapass Pungboon, Captain, RTN, JIT's Software Development Division, Supreme Command Headquarters, Bangkok, Thailand and author, 9 April 1995.

INITIAL DISTRIBUTION LIST

	Address	No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2.	Library, Code 52 Naval Postgraduate School Monterey, California 93943-5101	2
3.	Professor William Gates, Code SM/Gt Department of Systems Management Naval Postgraduate School Monterey, California 93943-5002	1
4.	Professor Lee Edwards, Code SM/Ed Department of Systems Management Naval Postgraduate School Monterey, California 93943-5002	1
5.	DR Supachai Sirayanone, Code MR/Sy Department of Meteorology Naval Postgraduate School Monterey, California 93943	1
6.	Office of The Military Attache, Royal Thai Embassy 2440 Foxhall Road, N.W. Washington, D.C. 20007	1
7.	MAJ Jitti Thongmuang 367 Army Quartermaster School Pracharat 1 Road Bangsue, Bangkok 10800 Thailand	1